APPENDIX A

PDR SOFTWARE, SOFTWARE SUPPORT, AND QUALIFICATION PROVISIONS

APPENDIX A

PDR SOFTWARE, SOFTWARE SUPPORT, AND QUALIFICATION PROVISIONS

The following outline presents the top level outline for the PDR software, software support, and qualification provisions.

PDR Maintenance Outline

1. Software

   a. Source Files
   b. Executable Software
   c. Segmentation Scripts

2. “ As Built” Software Design

   a. Design Decisions
   b. Architecture
   c. Software Component Interaction
   d. Environment Variables
   e. Database Variables
   f. External System Interfaces

3. Compilation/Build Procedures

4. Modification Procedures

5. Qualification Provisions

A.1 Software

The following sections list the PDR software, executables, and segmentation scripts.

A.1.1 Source Files

This section is organized to present file names and short descriptions for the following types of source files:

a. Shell Scripts,
b. SQL Scripts,
c. Report Definition files,
d. Gain Momentum Interchange Format Files, and
e. Data and text files used by the PDR application.

A.1.1.1 Shell Scripts

The alphabetically ordered list of shell scripts and short descriptions are as follows:

File Names	Description
/h/PDR/Scripts/PDR_launch	PDR Initiation
/h/PDR/Scripts/check_active_gain_processes.csh	Within Limit of Gain Processes Check
/h/PDR/Scripts/check_active_gain_processes.tk	Pop-up with Excess Gain Processes Message
/h/PDR/Scripts/clean_up_collection_table	OPLAN-based Report Record Deletion Cleanup Utility
/h/PDR/Scripts/clean_up_geo_tucha_collection_table	GEO/TUCHA Report Record Deletion Cleanup Utility
/h/PDR/Scripts/copy_rda_collection	RDA Marked Data Duplication
/h/PDR/Scripts/delete_geo_tucha_table_entries	GEO/TUCHA Report Completion Record Deletion
/h/PDR/Scripts/delete_table_entries	OPLAN-Based Report Completion Record Deletion
/h/PDR/Scripts/get_current_printer.csh	Identification of Available Printers
/h/PDR/Scripts/get_current_printer_wish.csh	Identification of User’s Printer Choice
/h/PDR/Scripts/get_current_printer_wish.tk	Window of Printer Choices (TCL/TK)
/h/PDR/Scripts/get_current_printer_xterm.csh	Window of Printer Choices (xterm)
/h/PDR/Scripts/kill_processes	Termination of Unix Processes
/h/PDR/Scripts/launch_PDR	Start PDR Application
/h/PDR/Scripts/overwrite	File Contents Replacement
/h/PDR/Scripts/pdr_check_user_role	User PDR Access Verification
/h/PDR/Scripts/pdr_nohup	Child Process Initiation Without Parent Termination
/h/PDR/Scripts/pdr_setup_printer	Current Printer Configuration Parameters
/h/PDR/Scripts/replace	Text Substitution
/h/PDR/Scripts/report_monitor	Monitor Report Generation Process
/h/PDR/Scripts/report_monitor_window	Monitor Report Generation Icon
/h/PDR/Scripts/run_BG_report	BG Report Initiation
/h/PDR/Scripts/run_BH_report	BH Report Initiation
/h/PDR/Scripts/run_BI_report	BI Report Initiation
/h/PDR/Scripts/run_BJ_report	BJ Report Initiation
/h/PDR/Scripts/run_D3_report	D3 Report Initiation
/h/PDR/Scripts/run_F12A_report	F12A Report Initiation
/h/PDR/Scripts/run_F30_report	F30 Report Initiation
/h/PDR/Scripts/run_F51_F52_report	F51/F52 Report Initiation
/h/PDR/Scripts/run_FF_FE_report	FF/FE Report Initiation
/h/PDR/Scripts/run_report	Report Generation Process Control
/h/PDR/Scripts/run_simple_report	Default Report Generation Command
/h/PDR/Scripts/set_printer_flag	Site Printer-Present Indicator
/h/PDR/Scripts/start_bg_process.csh	Background Report Process Control
/h/PDR/Scripts/start_pdr	PDR Gain Application Startup
/h/PDR/Scripts/start_report	Report Generation Control Process
/h/PDR/Scripts/start_reportwriter	Invocation of ORACLE Reports

A.1.1.2 SQL Scripts

The alphabetically ordered list of SQL scripts and short descriptions is as follows:

File Names	Description
/h/PDR/sql/change_time.sql	Time Format Conversion
/h/PDR/sql/copy_rda_collection_tbl.sql	RDA Marked Data Set Duplication
/h/PDR/sql/pdr_bg_air_data.sql	Intermediate BG Air Data Creation
/h/PDR/sql/pdr_bg_amc_data.sql	Intermediate BG AMC Data Creation
/h/PDR/sql/pdr_bg_msc_data.sql	Intermediate BG MSC Data Creation
/h/PDR/sql/pdr_bg_sea_data.sql	Intermediate BG Sea Data Creation
/h/PDR/sql/pdr_f11e_sq_data.sql	Intermediate F11E_SQ Data Creation
/h/PDR/sql/pdr_f11e_sq_delete.sql	Intermediate F11E_SQ Data Deletion
/h/PDR/sql/pdr_f11e_tn_data.sql	Intermediate F11E_TN Data Creation
/h/PDR/sql/pdr_f11e_tn_delete.sql	Intermediate F11E_TN Data Deletion
/h/PDR/sql/pdr_bg_f11w_data.sql	Intermediate F11W Data Creation
/h/PDR/sql/pdr_bg_f11w_delete.sql	Intermediate F11W Data Deletion
/h/PDR/sql/populate_pdr_rqmts.sql	BI/BJ Requirements Insertion
/h/PDR/sql/pdr_f30_data.sql	Intermediate F30 Data Creation
/h/PDR/sql/pdr_f30_delete.sql	Intermediate F30 Data Deletion
/h/PDR/sql/pdr_fm_delete.sql	Intermediate FM Data Deletion
/h/PDR/sql/pdr_fm_rpts_data.sql	Intermediate FM Data Creation
/h/PDR/sql/remove_geo_tucha_tbl_entries.sql	GEO/TUCHA Records PDR Collection Tables Deletion
/h/PDR/sql/remove_tbl_entries.sql	PDR Collection Tables Deletion
/h/PDR/sql/run_pdr_errors.sql	BI/BJ Error Table Population

A.1.1.3 Report Definition Files

The alphabetically ordered list of Report Definition files and short descriptions follow below. **Note:** For each report, there are two matching files with a *.rep* extension. One file is the runtime version of the report for Sun platforms, the other is the runtime version for Hewlett-Packard platforms.

File Names	Description
/h/PDR/data/BG_AIR.rdf	BG Airlift Report File
/h/PDR/data/BG_AMC.rdf	BG AMC Report File
/h/PDR/data/BG_MSC.rdf	BG MSC Report File
/h/PDR/data/BG_SEA.rdf	BG Sealift Report File
/h/PDR/data/BI_F50.rdf	OPLAN Logical Errors Report File
/h/PDR/data/BJ.rdf	Transportation Pre-Edit Report File
/h/PDR/data/F11D.rdf	Force List/Movement Requirements Working Paper Report File
/h/PDR/data/F11E_SQ.rdf	Time-Phased Transportation Rqmts Working Paper Square Footage Report File
/h/PDR/data/F11E_TN.rdf	Time-Phased Transportation Rqmts Working Paper Tonnage Report File
/h/PDR/data/F11W.rdf	Force Requirements Detail Report File

/h/PDR/data/F12A.rdf	Reference File Status Report
/h/PDR/data/F30.rdf	Transportation Summary Report
/h/PDR/data/F52.rdf	OPLAN (TPFDD) Compare Report File
/h/PDR/data/FE.rdf	Geolocation Report File
/h/PDR/data/FF.rdf	TUCHA Summary Report File
/h/PDR/data/FM_F11.rdf	FM Report File
/h/PDR/data/FM_RQT.rdf	FM Plan Reference Report File
/h/PDR/data/FM_RU.rdf	FM Roll up Report File

A.1.1.4 Gain Momentum Interchange Format File

PDR uses the Gain Momentum development tool to build the PDR graphical user interface (GUI). Because Gain Momentum stores all software components in an object-oriented database, the PDR user interface components can only be viewed and/or modified in the Gain Momentum environment. The PDR Gain Momentum components have been delivered in the form of textual files that were exported from the development Gain Momentum database. Gain Momentum refers to the export files as Interchange Format files. The following Gain Momentum Interchange Format file is required to build the PDR user interface in a Gain Momentum database:

File Name	Description
/h/PDR/gainif/PDRfolder.if	Predefined Reports library of Gain applications

A.1.1.5 Data and Text Graphics Files

PDR uses data and text files in support of multiple runtime functions.

A.1.1.5.1 Help Files

The online help capabilities of the PDR application consist of the following text files. The first file identifies the links between the Gain help engine and the specific PDR applications

File Names	Description
/h/PDR/src/helpdata/help.config	Online help links definition
/h/PDR/src/helpdata/geo_select.txt	Online help text for the GEO select screen
/h/PDR/src/helpdata/pdr_main.txt	Online help text for the OPLAN select screen
/h/PDR/src/helpdata/pdr_selection.txt	Online help text for the PDRRunReport screen
/h/PDR/src/helpdata/select.txt	Online help text for the Tucha select screen
/h/PDR/src/helpdata/select_sort.txt	Online help text for Sort Options selection screen
/h/PDR/src/helpdata/tucha.txt	Online help text for the Tucha criteria screen

A.1.1.5.2 Text Files

The text files shown below are used in reporting the termination status of a background report.

File Names	Description
/h/PDR/data/errors.txt	Error message of a background invocation of ORACLE Reports
/h/PDR/data/finished.txt	Termination status of an invocation of ORACLE Reports

A.1.2 Executable Software

An executable PDR user interface application consists of an executable file named *gain.exe*, application library files that have an extension of *.glb*, and standard Gain Momentum libraries. The application library files are listed below. The PDR software package incorporates a number of RDA applications; consequently, the list includes RDA library files in addition to the PDR library file.

a. */h/PDR/pdr_home/PDRdist/data/gccsutilities.glb*,
b. */h/PDR/pdr_home/PDRdist/data/predefinedreports.glb*,
c. */h/PDR/pdr_home/PDRdist/data/rdacargo.glb*,
d. */h/PDR/pdr_home/PDRdist/data/rdadevtools.glb*,
e. */h/PDR/pdr_home/PDRdist/data/rdaforcemodules.glb*,
f. */h/PDR/pdr_home/PDRdist/data/rdamain.glb*,
g. */h/PDR/pdr_home/PDRdist/data/rdamapping.glb*,
h. */h/PDR/pdr_home/PDRdist/data/rdamergecompare.glb*,
I. */h/PDR/pdr_home/PDRdist/data/rdareferencefiles.glb*,
j. */h/PDR/pdr_home/PDRdist/data/rdarequirements.glb*,
k. */h/PDR/pdr_home/PDRdist/data/rdart.glb*,
l. */h/PDR/pdr_home/PDRdist/data/rdaselect.glb*,
m. */h/PDR/pdr_home/PDRdist/data/rdatimeline.glb*,
n. */h/PDR/pdr_home/PDRdist/data/rdautility.glb*, and
o. */h/PDR/pdr_home/PDRdist/data/system.glb*.

Gain Momentum refers to an executable application that is built with Gain Momentum tools as a runtime.

A.1.3 Segmentation Scripts

The PDR segmentation process followed the GCCS Integration Standard (see Section 2.0, Reference *l*.). The PDR application is divided into two segments. The PDRSRV segment installs the ORACLE database portion of the PDR application onto the server machine. The PDR segment loads the PDR application onto the client machine. For more information on the PDR segmentation process, refer to the Installation Instruction Input for Predefined Reports (see Section 2.0, Reference *d*.).

A.1.3.1 PDRSRV Segment Files

PDRSRV has the following five sets of files:

a. PDRSRV Characteristics, which uniquely identifies the hardware and software characteristics of the specific ORACLE version used by the PDR application as specified in the corresponding PDR segment.

b. Post Install, which installs the PDR database server segment on a GCCS database server.

c. Deinstall, which removes the PDR database objects from a GCCS database server.

d. Enable a new PDR User, which adds a new PDR user to the list of authorized users.

e. Disable a PDR User, which removes a PDR user from the list of authorized users.

The files in each set are listed below. The filename is shown in parentheses.

a. PDRSRV Characteristics. The files comprising this set are located in the */h/PDRSRV/SegDescrip* directory.

   1. COTS List (*Requires*),
   2. Hardware (*Hardware*),
   3. Model Name (*ModName*),
   4. Release Notes (*ReleaseNotes*),
   5. Security (*Security*),
   6. Segment Type (*SegType),*
   7. PDRSRV Version (*VERSION*), and
   8. PDRSRV Validation Stamp (*Validated*).

b. Post Install. The files comprising this set are located in three directories. The *PostInstall* file is located in the */h/PDRSRV/SegDescrip* directory; the files having a *.sql* extension are located in the */h/PDRSRV/sql* directory. The other files are located in the */h/PDRSRV/Scripts* directory.

   1. Post Install PDR (*PostInstall*),
   2. PDR Set Directory (*PostInstall.set_dir*),
   3. ORACLE Password Determination (*determine_oracle_password*),
   4. ORACLE Password Setting (*enter_new_oracle_password*),
   5. JOPES_REPORTS User Account Creation (*pdr_cr_jr_user.sql*),
   6. PDR JOPES_REPORTS Package Creation (*pdr_cr_packages_jr.sql*),
   7. PDR TABLE_MASTER Package Creation (*pdr_cr_packages_tm.sql*),
   8. PDR Role Creation (*pdr_cr_role.sql*),
   9. PDR Report Options Table Creation (*pdr_cr_table_report_options.sql*),
   10. PDR Database Tables Creation (*pdr_cr_tables.sql*),
   11. BG Reports Tables Creation (*pdr_cr_tables_bg.sql*),
   12. PDR Database Collection Tables Creation (*pdr_cr_tables_coll.sql*),
   13. F11E_SQ Report Table Creation (*pdr_cr_tables_f11e_sq.sql*),

14. F11E_TN Report Table Creation (*pdr_cr_tables_f11e_tn.sql*),
15. PDR F11W Intermediate Table Creation (*pdr_cr_tables_f11w.sql*),
16. F30 Report Tables Creation (*pdr_cr_tables_f30.sql*),
17. FM Reports Tables Creation (*pdr_cr_tables_fm.sql*),
18. PDR Database Selection Criteria Tables Creation (*pdr_cr_tables_selection.sql*),
19. PDR Database Sort Tables Creation (*pdr_cr_tables_sort.sql*),
20. PDR Database System Tables Creation (*pdr_cr_tables_sys.sql*),
21. PDR Corrective Action Table (*pdr_cr_tables_t28.sql*),
22. PDR Tablespace Creation (*pdr_cr_tablespace.sql*),
23. PDR F11E Report View Creation (*pdr_cr_view_f11e_sq.sql*),
24. PDR F11E Report View Creation (*pdr_cr_view_f11e_tn.sql*),
25. PDR Views Creation (*pdr_cr_views.sql*).
26. PDR Synonyms Deletion (*pdr_drop_all_syn.sql*),
27. Extermination of PDR User (*pdr_drop_jr_user.sql*),
28. Complete Removal of PDR (*pdr_full_deinstall.sql*),
29. PDR Creation of User Synonyms (*pdr_get_user_info.sql*),
30. PDR Database Permissions (*pdr_grant_jr.sql*),
31. PDR Database Permissions To TableMaster Objects (*pdr_grant_tm.sql*),
32. RDA Database Errors Handler (*pdr_insert_logical_errors.sql*),
33. PDR Sort Rows Insertion (*pdr_insert_sort_columns.sql*),
34. PDR Standard Sort Options (*pdr_insert_sort_criteria.sql*),
35. PDR Server Partial Deinstallation (*pdr_partial_deinstall.sql*),
36. BG_AIR Report Package Definition (*pdr_pk_bg_air_body.sql*),
37. BG_AIR Report Package Specification (*pdr_pk_bg_air_spec.sql*),
38. BG_AMC Report Package Definition (*pdr_pk_bg_amc_body.sql*),
39. BG_AMC Report Package Specification (*pdr_pk_bg_amc_spec.sql*),
40. F11E_SQ Report Package Definition (*pdr_pk_f11e_sq_body.sql*),
41. F11E_SQ Report Package Specification (*pdr_pk_f11e_sq_spec.sql*),
42. F11E_TN Report Package Definition (*pdr_pk_f11e_tn_body.sql*),
43. F11E_TN Report Package Specification (*pdr_pk_f11e_tn_spec.sql*),
44. F11W Report Package Definition (*pdr_pk_f11w_body.sql*),
45. F11W Report Package Specification (*pdr_pk_f11w_spec.sql*),
46. F30 Report Package Definition (*pdr_pk_f30_body.sql*),
47. F30 Report Package Specification (*pdr_pk_f30_spec.sql*),
48. FM Reports Package Definition (*pdr_pk_fm1_body.sql*),
49. FM Reports Package Specification (*pdr_pk_fm1_spec.sql*),
50. FM Reports Package Definition (*pdr_pk_fm2_body.sql*),
51. FM Reports Package Specification (*pdr_pk_fm2_spec.sql*),
52. FM Reports Package Definition (*pdr_pk_fm3_body.sql*),
53. FM Reports Package Specification (*pdr_pk_fm3_spec.sql*),
54. FM Reports Package Definition (*pdr_pk_fm4_body.sql*),
55. FM Reports Package Specification (*pdr_pk_fm4_spec.sql*),
56. FM Reports Package Definition (*pdr_pk_fm5_body.sql*),
57. FM Reports Package Specification (*pdr_pk_fm5_spec.sql*),
58. FM Reports Package Definition (*pdr_pk_fm_body.sql*),
59. FM Reports Package Specification (*pdr_pk_fm_spec.sql*),
60. PDR Synonym Package Bodies (*pdr_pk_syn_body.sql*),
61. PDR Synonym Package Specifications (*pdr_pk_syn_spec.sql*),

62. BI/BJ Package Bodies (*pdr_pk_ve1_body.sql*),
63. BI/BJ Package Specifications (*pdr_pk_ve1_spec.sql*),
64. PDR Tables Row Insertion Control (*pdr_populate_tables.sql*),
65. Jopes_Reports Table Synonyms Creation (*pdr_run_updates.sql*),
66. Create PDR Synonyms (*pdr_update_syn.sql*),
67. Verify Tablespace Existence (*pdr_verify_tablespace.sql*),
68. Count of User Selected Records Definition (*rda_pk_ucp_counts_body.sql*), and
69. Count of User Selected Records Specification (*rda_pk_ucp_counts_spec.sql*).

c. Deinstall. The files comprising this set are located in three directories. The *Deinstall* file is located in the */h/PDRSRV/SegDescrip* directory; the files having a *.sql* extension are located in the */h/PDRSRV/sql* or in the */h/PDRSRV/install* directory. The *DEINSTALL.warning* file is located in the */h/PDRSRV/Scripts* directory.

1. Deinstall PDR (*DEINSTALL*),
2. Dropping PDR Tablespace Warning (*DEINSTALL.warning*),
3. Prompt for Full/Partial PDR Deinstallation (*pdr_deinstall_msg*),
4. PDR Full Deinstallation Process (*pdr_full_deinstall*),
5. PDR Identify User (*pdr_get_user_info.sql*)
6. PDR Full Database Deinstall Procedure (*pdr_full_deinstall.sql*),
7. PDR Partial Database Deinstall Procedure (*pdr_partial_deinstall.sql*),
8. PDR Synonyms Deletion (*pdr_drop_all_syn.sql*), and
9. PDR Stored Database Commands Execution (*pdr_run_updates.sql*).

d. Create a New PDR User. The UNIX script files comprising this set are located in the */h/PDRSRV/install* directory; the files having a *.sql* extension are located in the */h/PDRSRV/sql* or in the */h/PDRSRV/install* directory.

1. Enable PDR User (*pdr_enable_user.csh*),.
2. Verify PDR User (*pdr_check_user.sql*),
3. Assign PDR Role to User (*pdr_assign_to_role.sql*),
4. Verify ORACLE User Account Exists (*pdr_get_user_info.sql*),
5. Create PDR User Synonyms SQL (*pdr_cr_synonyms.sql*), and
6. Assign PDR Role to User (*pdr_run_updates.sql*).

e. Drop a PDR User. The UNIX script files comprising this set are also located in the */h/PDRSRV/install* directory; the files having a *.sql* extension are located in the */h/PDRSRV/sql* or the */h/PDRSRV/install* directory.

1. Disable PDR User (*pdr_disable_user.csh*),
2. Verify User Account (*pdr_check_user.sql*),
3. Drop PDR User's PDR Role (*pdr_revoke_user.sql*),
4. Get User Account Information (*pdr_get_user_info.sql*),
5. Create PDR SQL Drop Commands (*pdr_drop_synonyms.sql*), and
6. Execute PDR SQL Commands (*pdr_run_updates.sql*).

A.1.3.2 PDR Segment Files

PDR has the following three sets of files:

a. PDR Characteristics, which identifies hardware and software characteristics of a specific PDR release.

b. Post Install, which installs the PDR application on a GCCS platform.

c. Deinstall, which removes the PDR application from a GCCS platform.

The files in each set are listed below. The filename is shown in parentheses.

a. PDR Characteristics. The UNIX script files comprising this set are located in the */h/PDR/SegDescrip* directory.

   1. COTS List (*Requires*),
   2. Hardware (*Hardware*),
   3. Model Name (*ModName*),
   4. Release Notes (*ReleaseNotes*),
   5. Security (*Security*),
   6. Segment Type (*SegType*),
   7. PDR Version (*VERSION*), and
   8. Validation Stamp (*Validated*).

b. Post Install. The files comprising this set are located in two directories. The *PostInstall* file is located in the */h/PDR/SegDescrip* directory; the *set_printer_flag* is located in the */h/PDR/scripts* directory.

   1. PostInstall PDR (*PostInstall*), and
   2. Set Printer Availability Flag (*set_printer_flag*).

c. Deinstall. This file (*DEINSTALL*) is located in the */h/PDR/SegDescrip* directory.

A.2 "As Built" Software Design

The PDR application provides the user the capability to initiate a number of JDS and JOPS OPLAN-based reports whose formats are predefined. PDR also provides the capability to display standard reference file data, and optionally to generate a hard-copy report. The reference file display/print capability is currently limited to the GEO and TUCHA data reference files, and the Reference File Summary Report. The PDR application may be invoked directly from JNAV or indirectly through RDA. **Note:** Only the OPLAN-based reports are available from RDA.

A.2.1 Design Decisions

This section describes the significant design decisions made during the development of the PDR application.

A.2.1.1 RDA "Look and Feel"

The users expressed the need for a consistent " look and feel" for all the GCCS products. When the development of the PDR application began, the RDA development was well under way, and its look-and-feel was already accepted by the users. A common look and feel with RDA is especially important to PDR because the OPLAN-based reporting capabilities of PDR can be invoked from RDA by the user in a seamless fashion (that is, it is not apparent to the user that another application has started). The decisions were made to use Gain Momentum, the development tool used by RDA, and to design similar windows for PDR.

A.2.1.2 Software Reuse

The PDR application has made use of existing code to limit the extent of new development whenever possible. Many functions that the PDR system has to perform also exist in RDA. Since the development of RDA preceded the development of PDR, these functions had already been created by RDA. Instead of writing them again for PDR, the decision was made to link the PDR application to the functions in RDA. These functions fall in the areas of connections to the ORACLE database, user account identification, data identification for OPLAN-based report generation, and error handling. The RDA Gain Momentum applications listed below are incorporated into the PDR runtime when it is generated.

a. gccsutilities.glb,
b. rdacargo.glb,
c. rdadevtools.glb,
d. rdaforcemodules.glb,
e. rdamain.glb,
f. rdamapping.glb,
g. rdamergecompare.glb,
h. rdareferencefiles.glb,
I. rdarequirements.glb,
j. rdart.glb,
k. rdaselect.glb,
l. rdatimeline.glb, and
m. rdautility.glb.

Of these applications, PDR links directly to functions in the following: RDAMergeCompare, RDARequirements, RDASelect, and RDAUtility. The other applications are incorporated into the PDR runtime because they are invoked by the RDA applications to which PDR links.

A.2.1.3 Report Generation

The user has three choices for the destination of a generated report: 1) the window for report preview, 2) directly to a printer, or 3) a named file. When the report is executed for preview, it is run in foreground mode; when the destination is the printer or a file, the report is run in background mode and a message mailed to the user when the report is completed. **Note:** When the requested destination is a file, the format of the report file depends on the setting of the default ORACLE printer. When the default is a PostScript printer, the file content is also PostScript; otherwise, the content is American Standard Code for Information Interchange (ASCII). Experience to date has been that a PostScript printer must be online for the ORACLE Reports process to complete without error; this is the production mode. In development mode, the development team has been able to generate ASCII files without an online ASCII printer. See Paragraph A.2.3.2.3, ORACLE Reports.

A.2.1.4 Report Security Classification

When a report is initiated, the security classification of the output defaults to the security classification of the OPLAN for OPLAN-based reports or of the Reference File for RFP capability. The user has the option of adjusting the report security classification. The selected classification or the default, if not over-ridden by the user, is printed in the top and bottom margins on all pages of the report. Currently, the highest level of classification is Secret.

A.2.1.5 Collection Tables

The PDR application uses collection tables to identify the target set of records for inclusion on any given report. The PDR collection tables enable a user to request a report for printing or for output to a file, and immediately enter a different transaction against any other set of records, even in the same tables as the requested report. Reuse of the application or the database does not depend upon, or wait on, the completion of the requested report. The information stored in the collection tables is limited to information that identifies the record, e.g., OPLAN ID, requirement number, and requirement type, and does not include potentially sensitive OPLAN information. To support multiple concurrent users, or overlapping report generation by a single user, the collection table entries are further identified by the user name of the logged user and the date/time of the request for a report. The report generation software uses these variables to identify the correct set of information for any given reporting instance.

A.2.1.6 Report Formats

The format and presentation of the reports were developed in ORACLE Reports 2.0. At the request of the JOPES User Review Panel (URP), the format of the PDR reports corresponds closely to the format of the legacy reports that they replace. Information on the contents of each PDR report is found in the PDR Users Manual, refer to Section 2.0, item m. Reports in the initial PDR delivery included a logo on the header pages and the first data page. These were removed at the request of the JOPES URP.

A.2.2 Architecture

The PDR architecture can be described in hierarchical fashion. As with other GCCS JOPES applications, each element in the hierarchy consists of a set of physical files. The highest elements are termed work packages, which provide sets of related functionality. Work packages themselves consist of logical units, which are logical collections of physical files. Finally, each logical unit consists of one or more functional units.

A.2.2.1 Work Package Structure

The PDR application consists of a single work package, which is divided into the logical units shown below and in schematic form shown in Figure A.2.2.1-1, PDR Logical Units.

a. OPLAN-Based Reports,
b. GEO Paging/Reports,
c. TUCHA Paging/Reports,
d. Reference File Summary Report,
e. PDR Common Services, and
f. PDR Segmentation Services.

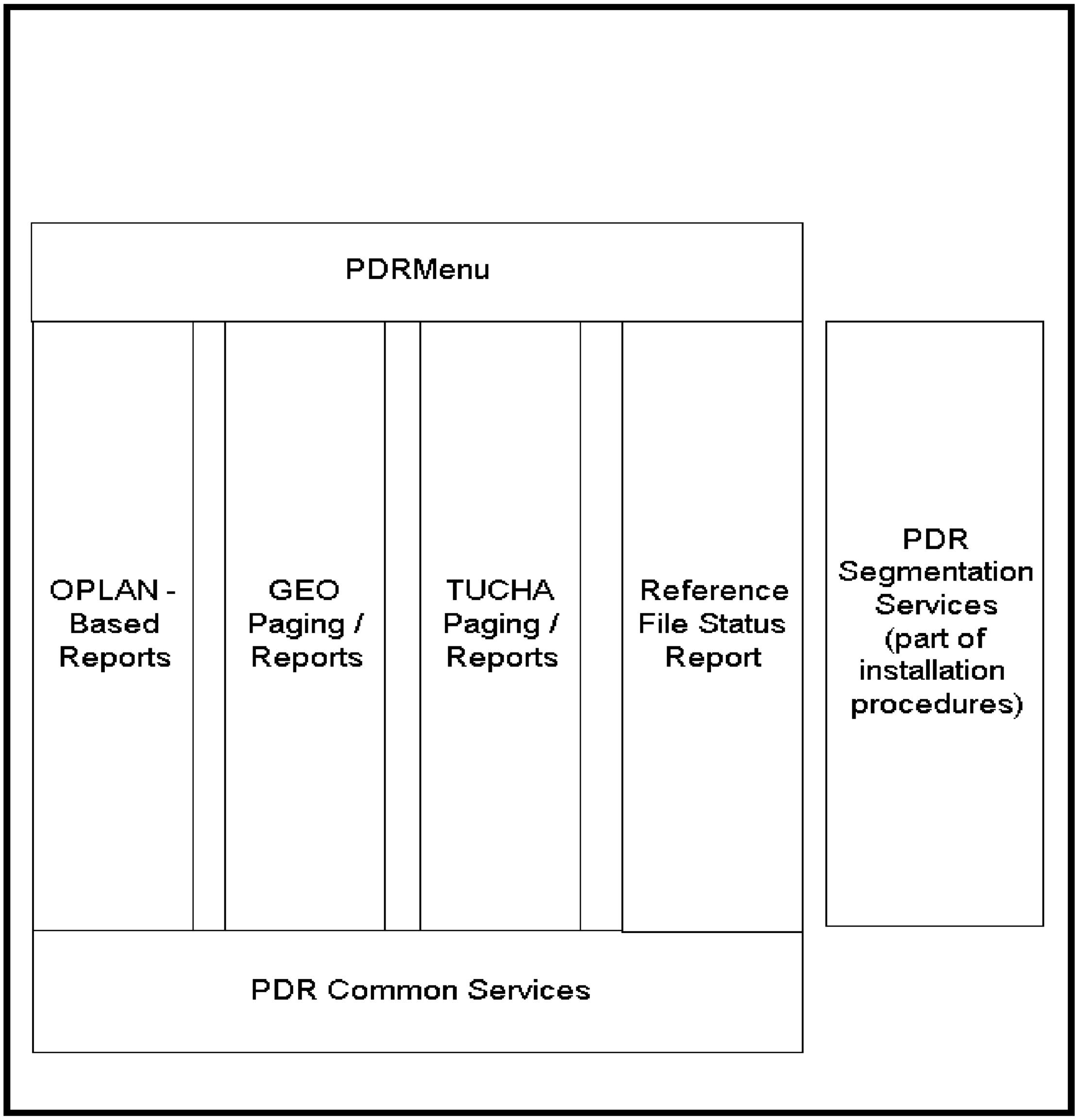

Figure A.2.2.1-1. PDR Logical Units

A.2.2.1.1 OPLAN-Based Reports

The OPLAN-based reports logical unit consists of the functional units necessary to support the generation of the reports listed below. In the list, the legacy report codes for JDS and JOPS are shown in parentheses after the report title.

a. Force Requirements Detail Report (BG/F11W),
b. Airlift Requirements Detail Report (BG),
c. AMC Requirements Detail Report (BG),
d. Sealift Requirements Detail Report (BG),
e. MSC Requirements Detail Report (BG),
f. Force List/Movement Requirements Working Paper (BH/F11D),
g. Time-Phased Transportation Requirements Working Paper - Tons (BH/F11E-TN),
h. Time-Phased Transportation Requirements Working Paper - SqFt (BH/F11E-SQ),
i. Plan Requirements Module Reference (D3),
j. Force Module Roll up Report (D3),
k. Force Module Report (D3),
l. Logical Errors Report (BI/F50),
m. Transportation Pre-Edit Report (BJ),
n. Transportation Summary Report (F30), and
o. OPLAN (TPFDD) Compare Report (F52).

A.2.2.1.2 GEO Paging/Reports

The GEO Paging/Reports logical unit consists of the functional units necessary to support the display and optional hardcopy reporting of selected GEO information. This is equivalent to the legacy FE (JDS) and F12E (JOPS) applications.

A.2.2.1.3 TUCHA Paging/Reports

The TUCHA Paging/Reports logical unit consists of the functional units necessary to support the display and optional hardcopy reporting of selected TUCHA information. This is equivalent to the legacy FF (JDS) and F12B (JOPS) applications.

A.2.2.1.4 Reference File Summary Report

The Reference File Status Report logical unit consists of the functional units necessary to support the display and optional hardcopy reporting of the Reference File Status Report. This is equivalent to the legacy F12A (JOPS) application.

A.2.2.1.5 PDR Common Services

The PDR Common Services logical unit consists of the functional units which are common to the functionally-oriented units (OPLAN, GEO, TUCHA, and Reference File Status). Included are the PDR application startup and shutdown services, high-level PDR navigation, ORACLE Reports interface, and other routines that are not specific to any particular functional unit.

A.2.2.1.6 PDR Segmentation Services

The PDR Segmentation Services logical unit consists of the script files used to install and deinstall the PDR application and to enable and disable a user for the PDR application.

A.2.2.2 Detailed Design

A.2.2.2.1 Logical and Functional Units

This paragraph describes the logical and functional units of the PDR work package. **Note:** In the Gain Momentum tool, which is used to build the PDR user interface, the basic work unit is an application. The PDR user interface is made up of PDR applications (described below) and RDA applications. When the PDR runtime is built, the RDA applications are incorporated into the resulting executable.

PDR can also be invoked from the RDA application or it can be invoked directly from the JNAV menu. When PDR is invoked from RDA, the flow of control is different than when PDR is invoked from JNAV. The initiation of PDR is discussed in Paragraph A.2.2.2.1.5.1, PDR Startup.

A.2.2.2.1.1 OPLAN-Based Reports

The OPLAN-based reports logical unit is composed of the functional units described in this paragraph.

A.2.2.2.1.1.1 User Interface

The PDR user interface for the OPLAN-based reports logical unit is implemented using a functional unit developed explicitly for PDR and incorporates a number of RDA applications.

A.2.2.2.1.1.1.1 PDR-Developed Functional Units

The "oplan_id" functional unit was developed to support the generation of OPLAN-based reports. This functional unit presents the user with a list of OPLANs for which the user is authorized, with the default OPLAN already highlighted.

Figure A.2.2.2.1.1.1.1-1, PDR Application Hierarchy, depicts the execution hierarchy of the PDR Gain Momentum applications. In the figure, a box represents a Gain Momentum application. Error-handling functions in the RDA Utility application are not included in this figure.

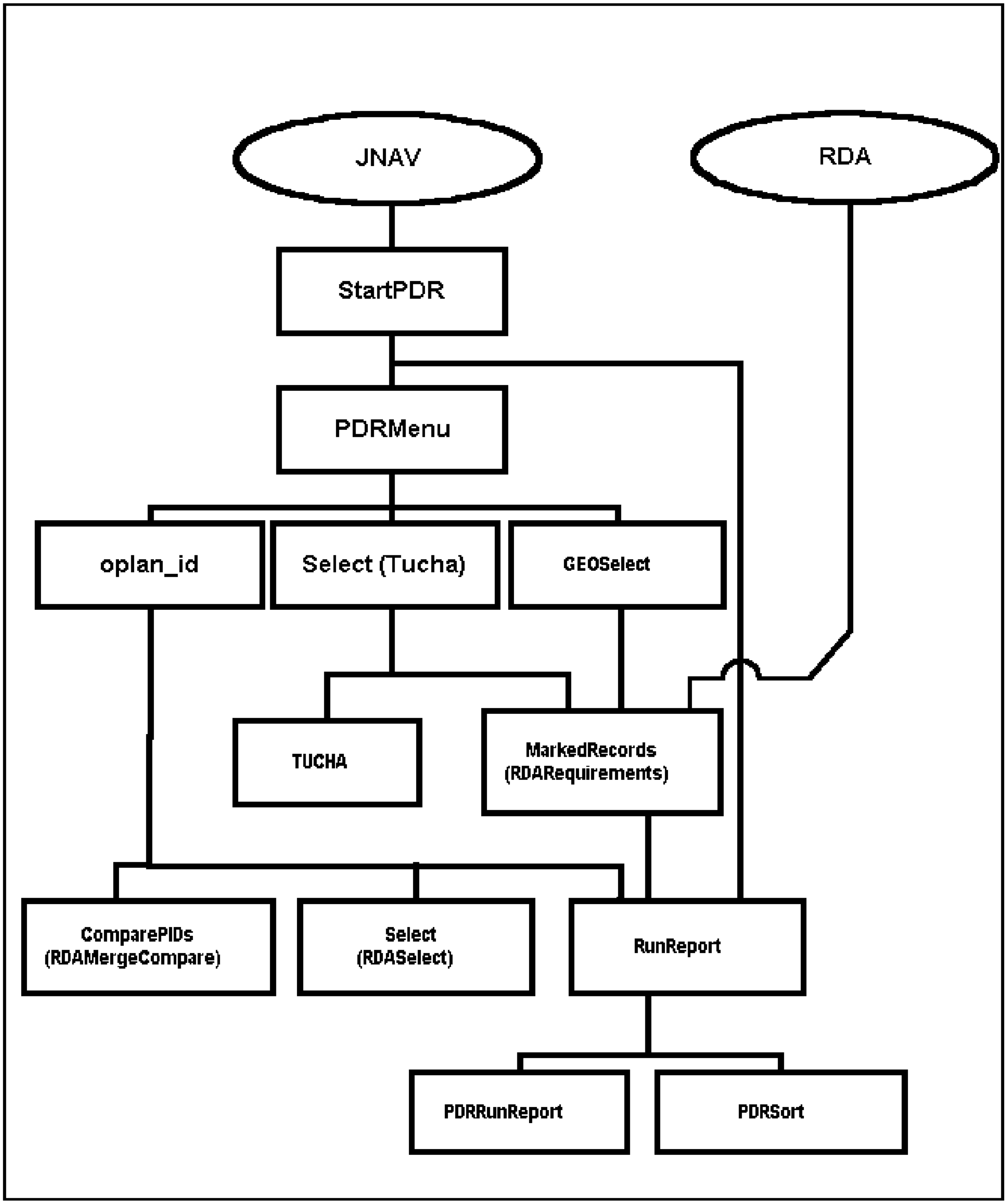

Figure A.2.2.2.1.1.1.1-1. PDR Application Hierarchy

A.2.2.2.1.1.1.2 RDA-Developed Functional Units

The functional units listed below are RDA applications that are invoked, directly or indirectly, by the PDR OPLAN-based reports functional unit. The current baseline versions of these RDA applications are incorporated into the PDR runtime when it is built. See Paragraph A.2.1.2, Software Reuse.

a. GCCSUtilities,
b. RDACargo,
c. RDADevTools,
d. RDAForceModules,
e. RDAMain,
f. RDAMapping,
g. RDAMergeCompare,
h. RDAReferenceFiles,
i. RDARequirements,
j. RDARt,
k. RDASelect,
l. RDATimeLine, and
m. RDAUtility.

A.2.2.2.1.1.2 Reports Interface

The interface between the PDR Gain application, where the user specifies and qualifies a report, and the report writer is accomplished using UNIX and SQL scripts. To achieve better performance, the BG, F11E, F11W, F30, and FM reports make use of interim tables, which are populated with report-specific and formatted data when the user initiates the report process. SQL scripts and PL/SQL packages of stored procedures that reside on the server are executed prior to the execution of the ORACLE Reports report generator for these reports. See Paragraph A.2.5.2.3.3, Intermediate Data Stores, for the scripts that create the interim tables. The following list shows the SQL scripts and PL/SQL packages that each report type uses. The client-based SQL scripts invoke the server-based stored procedures. Note that a stored procedure is composed of both a specification script that defines the procedures and their input and output parameters and a second script, which ORACLE refers to as the body, that contains the SQL and PL/SQL contents of each procedure.

The complexity of the data required for the F11E reports required special database functions in addition to the interim tables and report generators. For both versions of the F11E report, several views were created. See Paragraph A.2.5.2.4, Views, for the view creation scripts. The UNIX scripts are located in the */h/PDR/Scripts* directory. The SQL scripts are located in the */h/PDR/sql* directory and the PL/SQL scripts are located in the */h/PDRSRV/sql* directory. The following scripts are used:

a. UNIX Scripts
clean_up_collection_table
copy_rda_collection
delete_table_entries
run_BG_report
run_BH_report
run_BI_report
run_BJ_report

run_D3_report
run_FF_FE_report
run_F11W_report
run_F30_report
run_F51_F52_report

b. General SQL Scripts
change_time.sql
copy_rda_collection_tbl.sql
remove_tbl_entries.sql

c. Report-Specific SQL Scripts

Report	File
BG_AIR	*/h/PDR/sql/pdr_bg_air_data.sql* */h/PDR/sql/pdr_bg_air_delete.sql*
BG_AMC	*/h/PDR/sql/pdr_bg_amc_data.sql* */h/PDR/sql/pdr_bg_amc_delete.sql*
BG_SEA	*/h/PDR/sql/pdr_bg_sea_data.sql* */h/PDR/sql/pdr_bg_sea_delete.sql*
BG_MSC	*/h/PDR/sql/pdr_bg_sea_data.sql* */h/PDR/sql/pdr_bg_sea_delete.sql*
BI/BJ	*/h/PDR/sql/populate_pdr_rqmts.sql* */h/PDR/sql/run_pdr_errors.sql*
F11E-SQ	*/h/PDR/sql/pdr_f11e_sq_data.sql* */h/PDR/sql/pdr_f11e_sq_delete.sql*
F11E-TN	*/h/PDR/sql/pdr_f11e_tn_data.sql* */h/PDR/sql/pdr_f11e_tn_delete.sql*
F11W	*/h/PDR/sql/pdr_f11w_data.sql* */h/PDR/sql/pdr_f11w_delete.sql*
F30	*/h/PDR/sql/pdr_f30_data.sql* */h/PDR/sql/pdr_f30_delete.sql*
FM	*/h/PDR/sql/pdr_fm_rpts_data.sql* */h/PDR/sql/pdr_fm_delete.sql*

d. Report-Specific Server-Based Stored Procedure Scripts

Report	File
BG_AIR	*/h/PDRSRV/sql/pdr_pk_bg_air_body.sql* */h/PDRSRV/sql/pdr_pk_bg_air_spec.sql*
BG_AMC	*/h/PDRSRV/sql/pdr_pk_bg_amc_body.sql* */h/PDRSRV/sql/pdr_pk_bg_amc_spec.sql*
BI/BJ	*/h/PDRSRV/sql/pdr_pk_ve1_body.sql* */h/PDRSRV/sql/pdr_pk_ve1_spec.sql*
F11E-SQ	*/h/PDRSRV/sql/pdr_pk_f11e_sq_body.sql* */h/PDRSRV/sql/pdr_pk_f11e_sq_spec.sql*

F11E-TN	*/h/PDRSRV/sql/pdr_pk_f11e_tn_body.sql* */h/PDRSRV/sql/pdr_pk_f11e_tn_spec.sql*
F11W	*/h/PDRSRV/sql/pdr_pk_f11w_body.sql* */h/PDRSRV/sql/pdr_pk_f11w_spec.sql*
F30	*/h/PDRSRV/sql/pdr_pk_f30_body.sql* */h/PDRSRV/sql/pdr_pk_f30_spec.sql*
FM	*/h/PDRSRV/sql/pdr_pk_fm_body.sql* */h/PDRSRV/sql/pdr_pk_fm_spec.sql* */h/PDRSRV/sql/pdr_pk_fm1_body.sql* */h/PDRSRV/sql/pdr_pk_fm1_spec.sql* */h/PDRSRV/sql/pdr_pk_fm2_body.sql* */h/PDRSRV/sql/pdr_pk_fm2_spec.sql* */h/PDRSRV/sql/pdr_pk_fm3_body.sql* */h/PDRSRV/sql/pdr_pk_fm3_spec.sql* */h/PDRSRV/sql/pdr_pk_fm4_body.sql* */h/PDRSRV/sql/pdr_pk_fm4_spec.sql* */h/PDRSRV/sql/pdr_pk_fm5_body.sql* */h/PDRSRV/sql/pdr_pk_fm5_spec.sql*

Figures A.2.2.2.1.1.2-1 through A.2.2.2.1.1.2-12, as defined below, show the scripts' execution hierarchies for the OPLAN-based reports logical unit. **Note**: There are scripts shown in the figures that are not in the above list, e.g., *run_report*; there also are scripts that are subsidiary to the generation of a specific report that are not shown in the figures, e.g., *pdr_check_user_role.sql*. These scripts are covered in Paragraph 2.2.2.1.5, PDR Common Services. Figures A.2.2.2.1.1.2-9 and A.2.2.2.1.1.2-10 represent the execution hierarchies for multiple reports.

Figure	Description
A.2.2.2.1.1.2-1	Force Requirements Detail Report (BG/F11W) Script Execution Hierarchy
A.2.2.2.1.1.2-2	BG/Airlift Reports Script execution Hierarchy
A.2.2.2.1.1.2-3	BG/AMC Report Script Execution Hierarchy
A.2.2.2.1.1.2-4	BG/Sealift Report Execution Hierarchy
A.2.2.2.1.1.2-5	BG/M.C. Report Script Execution Hierarchy
A.2.2.2.1.1.2-6	Force List/Movement Requirements Working Paper (F11D) Script Execution Hierarchy
A.2.2.2.1.1.2-7	Time-Phased Transportation Reports Working Paper - Tonnage (F11E-TN) Script Execution Hierarchy
A.2.2.2.1.1.2-8	Time-Phased Transportation Requirements Working Paper - Square Footage (F11E-SQ) Script Execution Hierarchy
A.2.2.2.1.1.2-9	FM Reports Script Execution Hierarchy
A.2.2.2.1.1.2-10	Error Reports Script Execution Hierarchy
A.2.2.2.1.1.2-11	OPLAN (TPFDD) Comparison Report Script Execution Hierarchy
A.2.2.2.1.1.2-12	F30 Report Script Execution Hierarchy

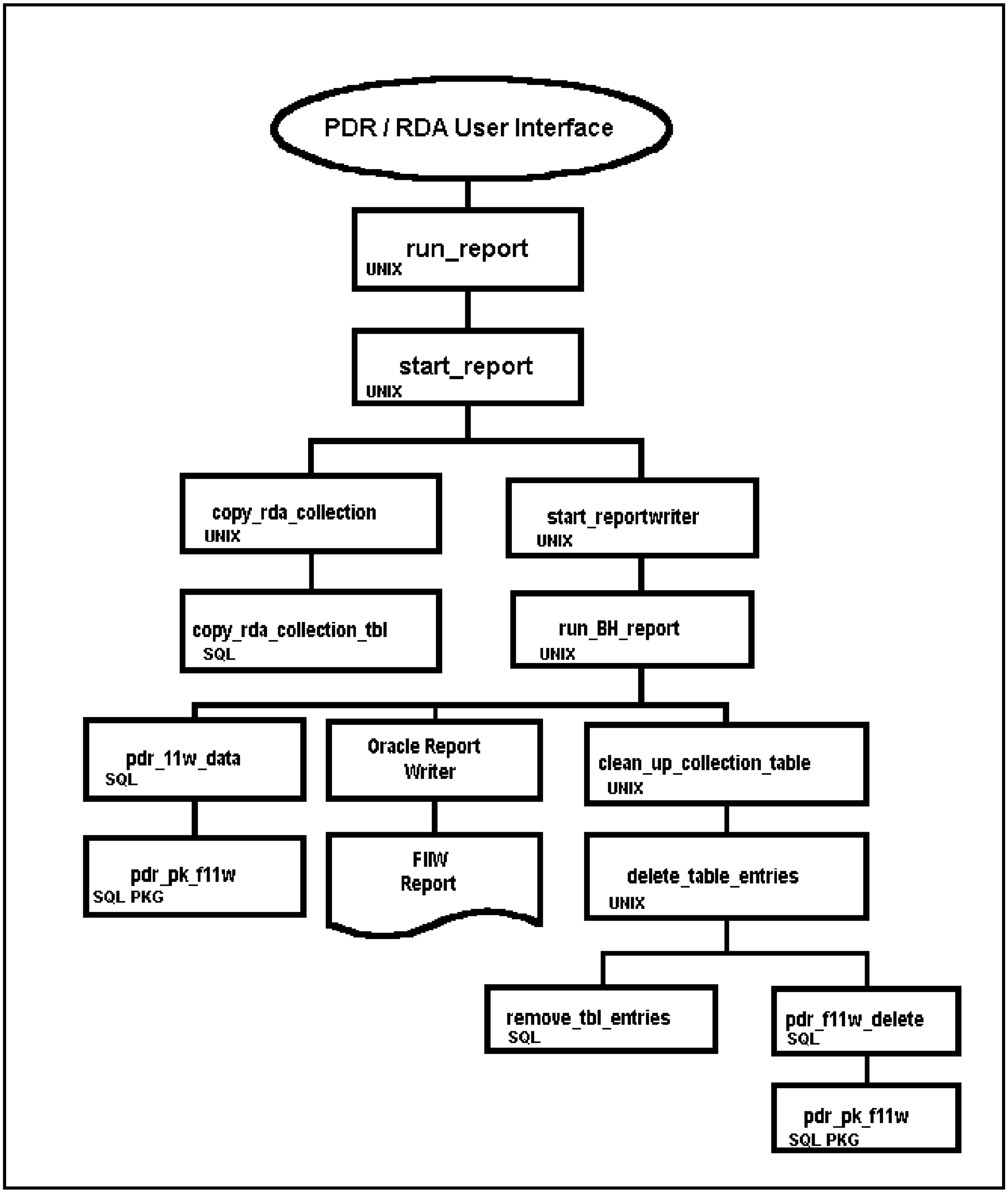

Figure A.2.2.2.1.1.2-1. Force Requirements Detail Report (BG/F11W)
Script Execution Hierarchy

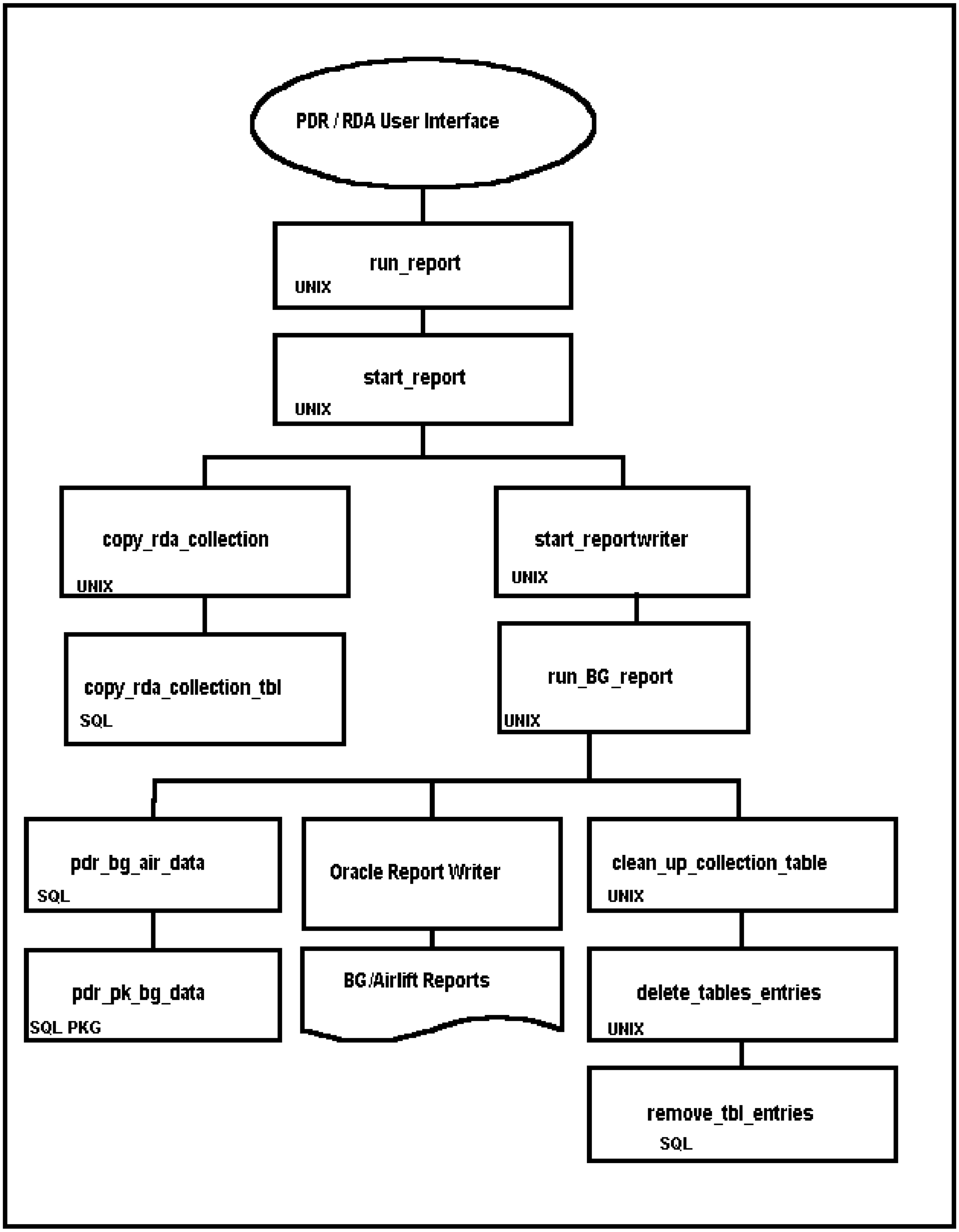

Figure A.2.2.2.1.1.2-2. BG/Airlift Reports Script Execution Hierarchy

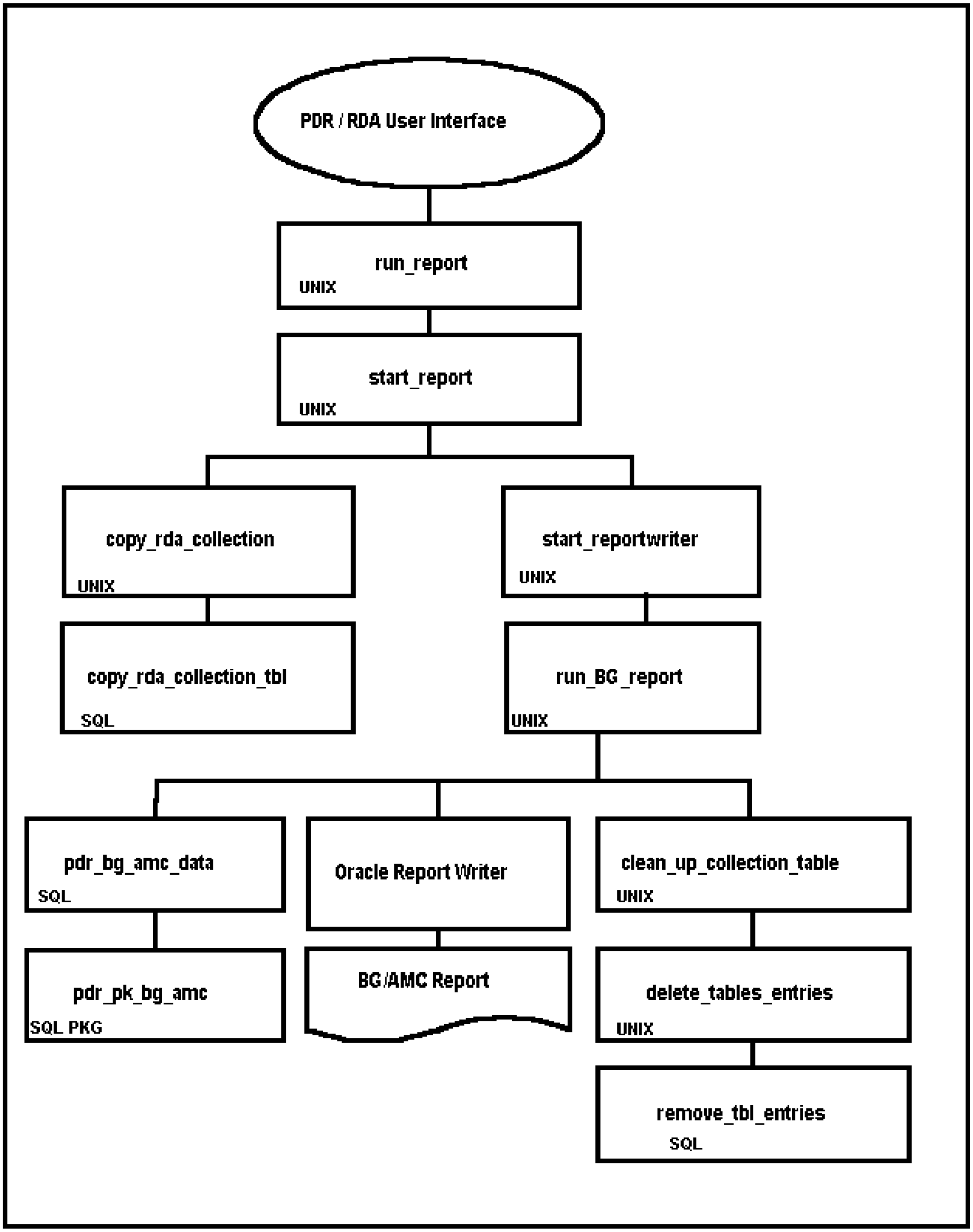

Figure A.2.2.2.1.1.2-3. BG/AMC Report Script Execution Hierarchy

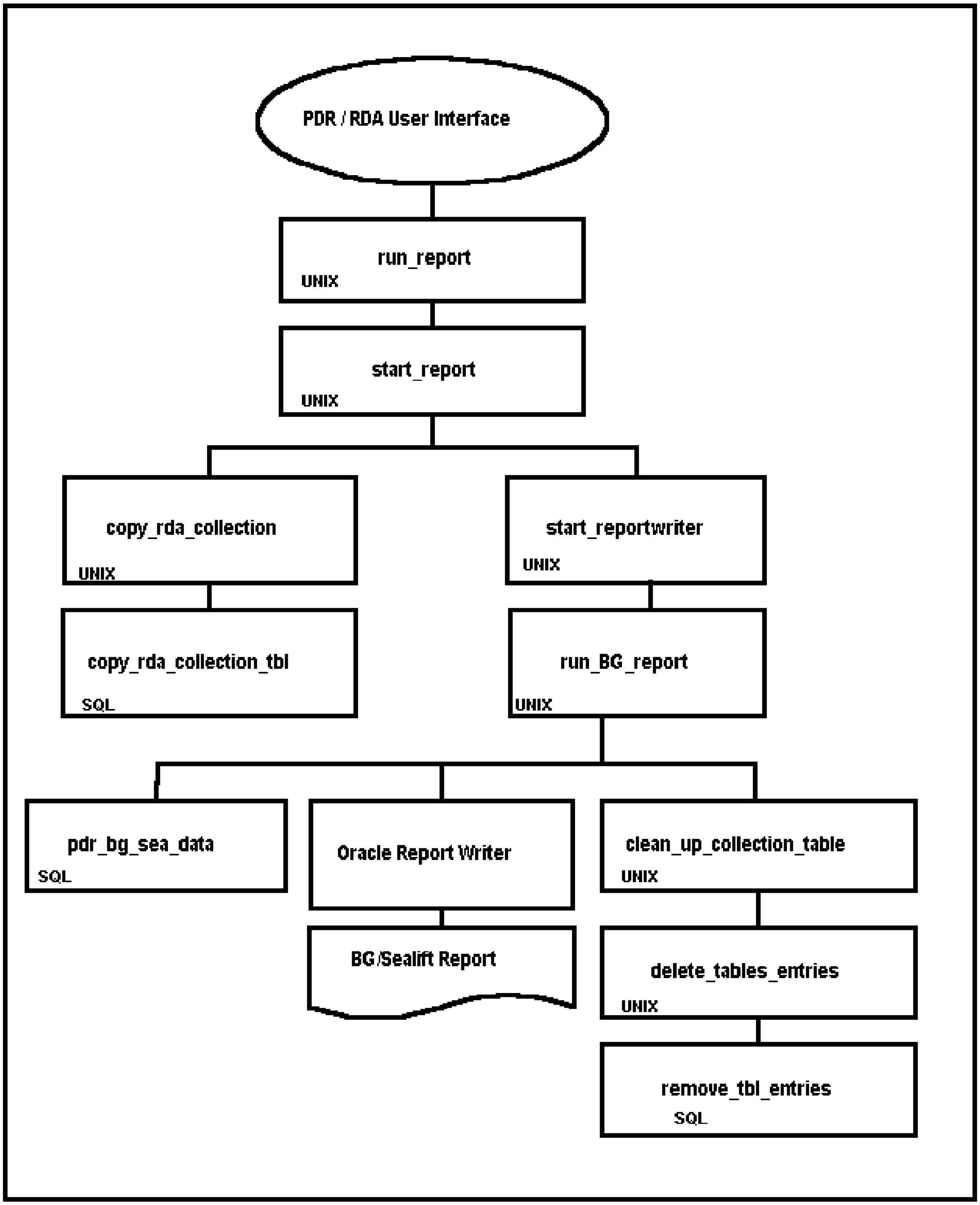

Figure A.2.2.2.1.1.2-4. BG/Sealift Report Execution Hierarchy

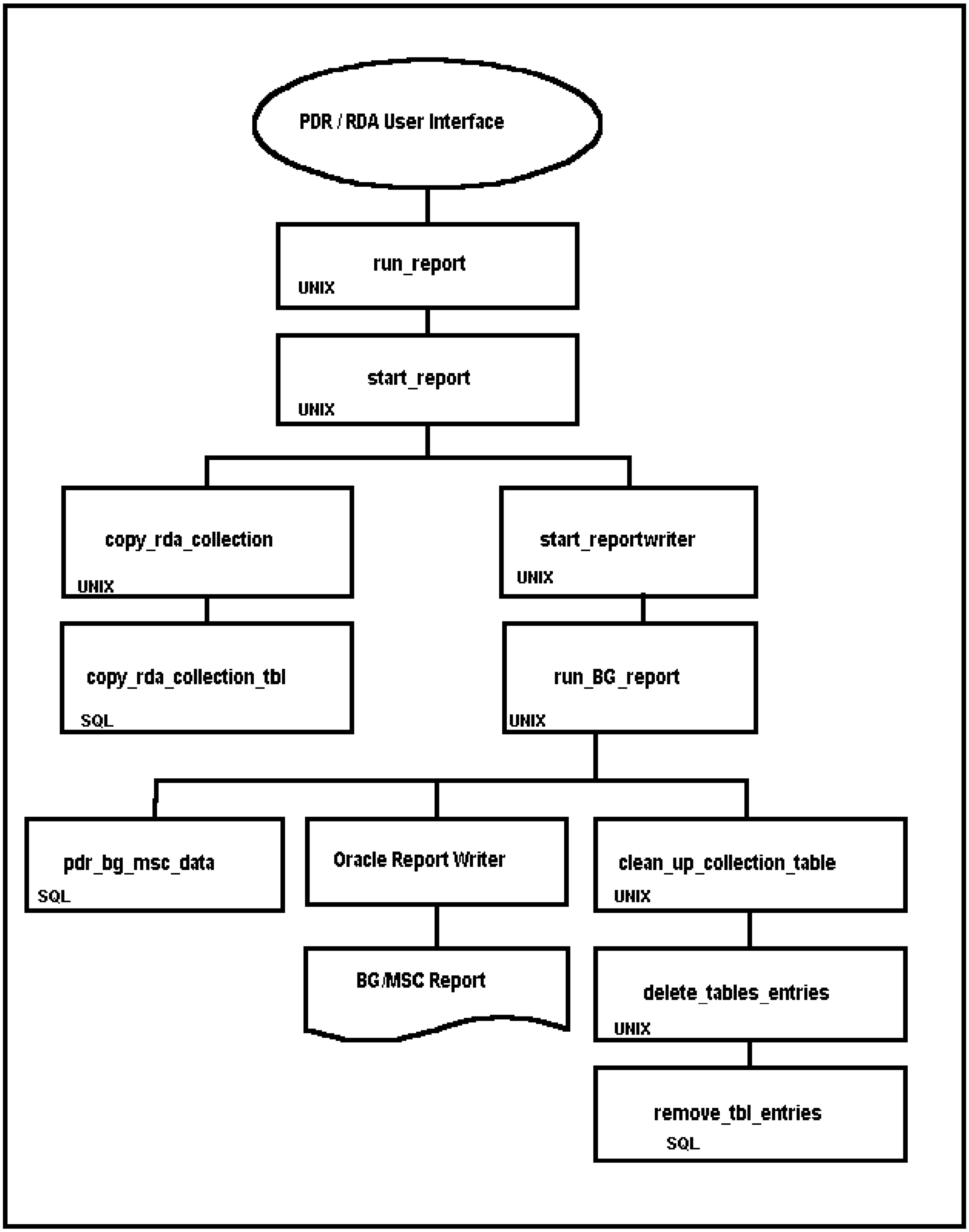

Figure A.2.2.2.1.1.2-5. BG/M.C. Report Script Execution Hierarchy

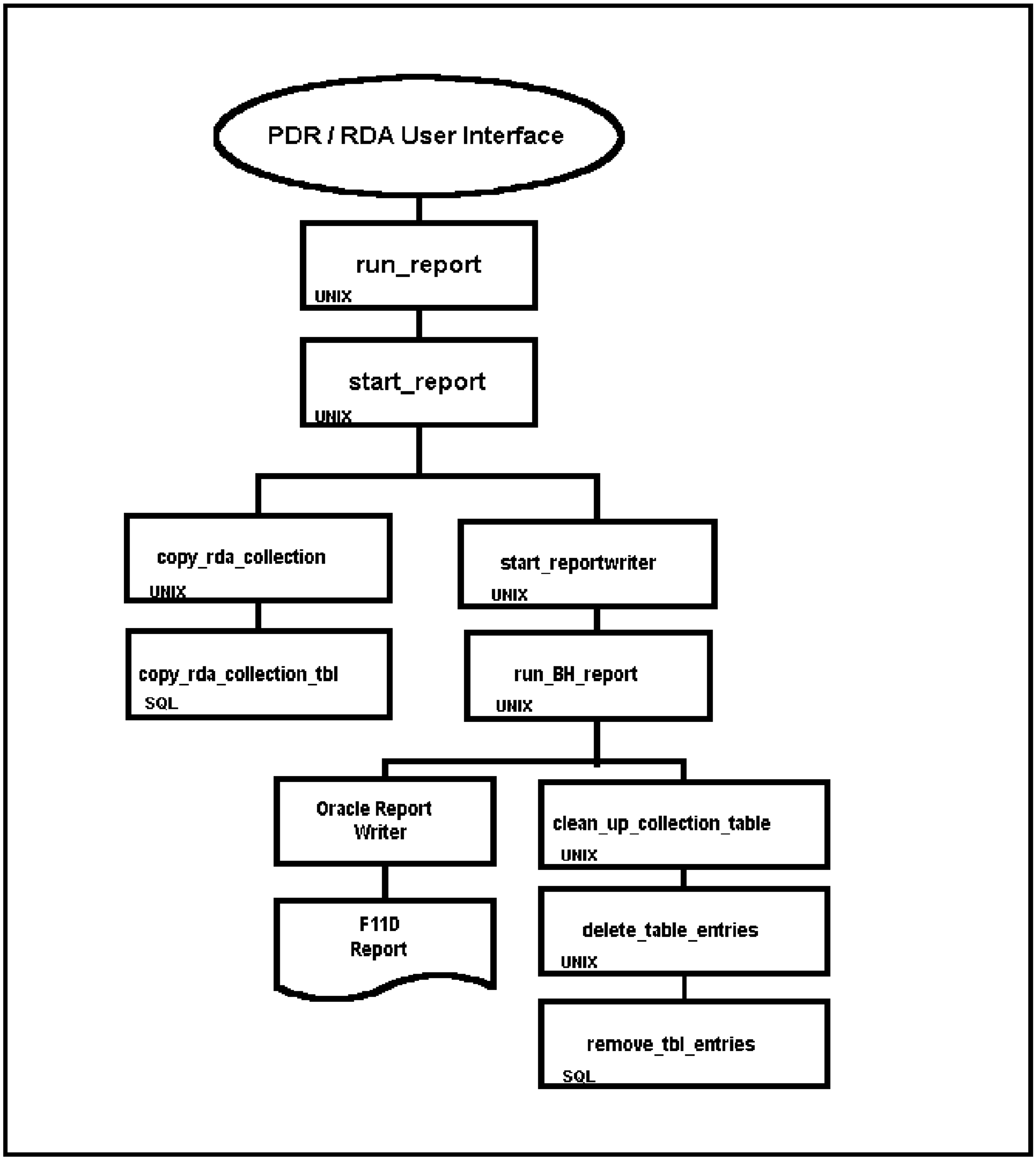

Figure A.2.2.2.1.1.2-6. Force List/Movement Requirements Working Paper (F11D) Script Execution Hierarchy

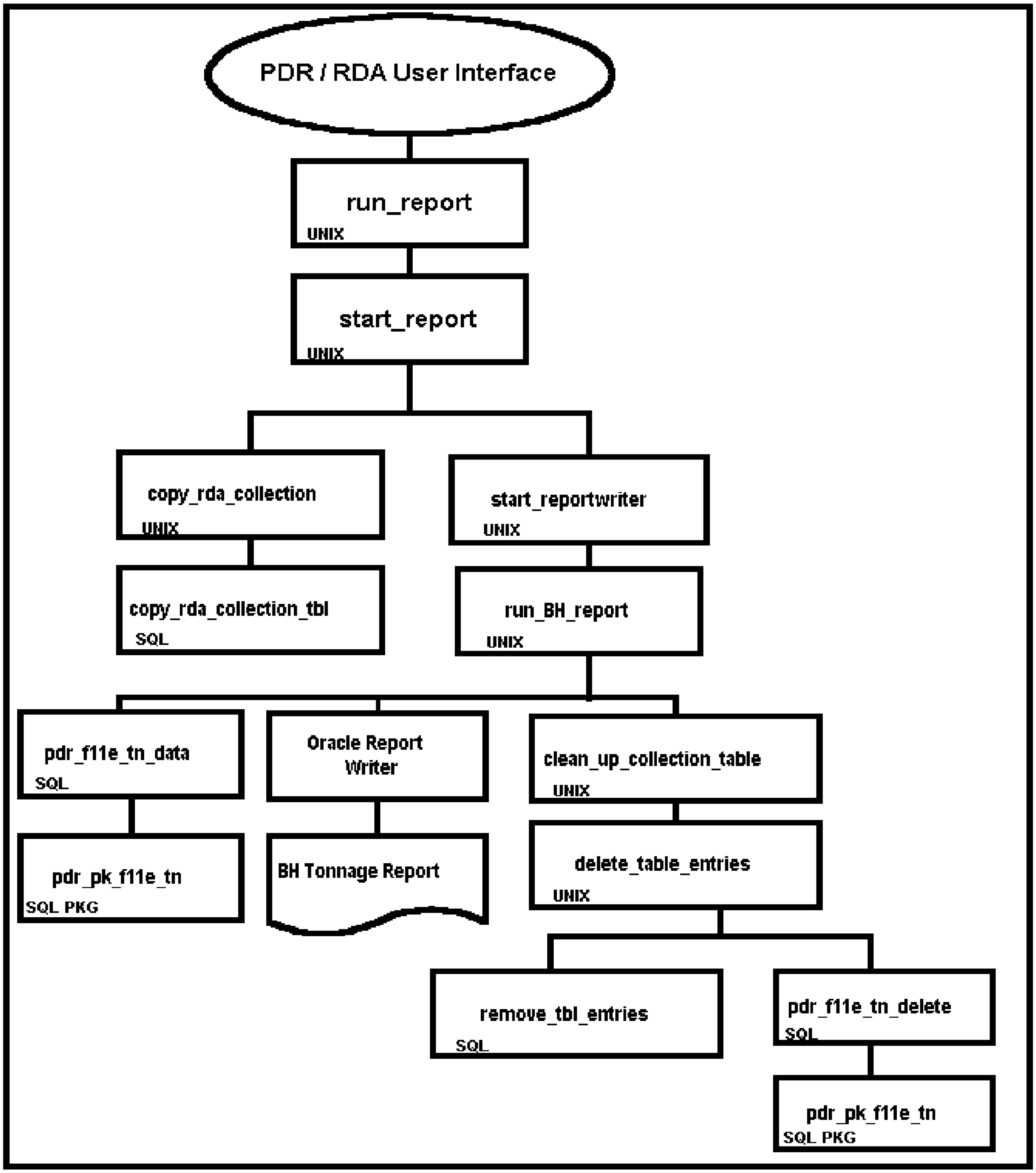

Figure A.2.2.2.1.1.2-7. Time-Phased Transportation Reports Working Paper - Tonnage (F11E-TN) Script Execution Hierarchy

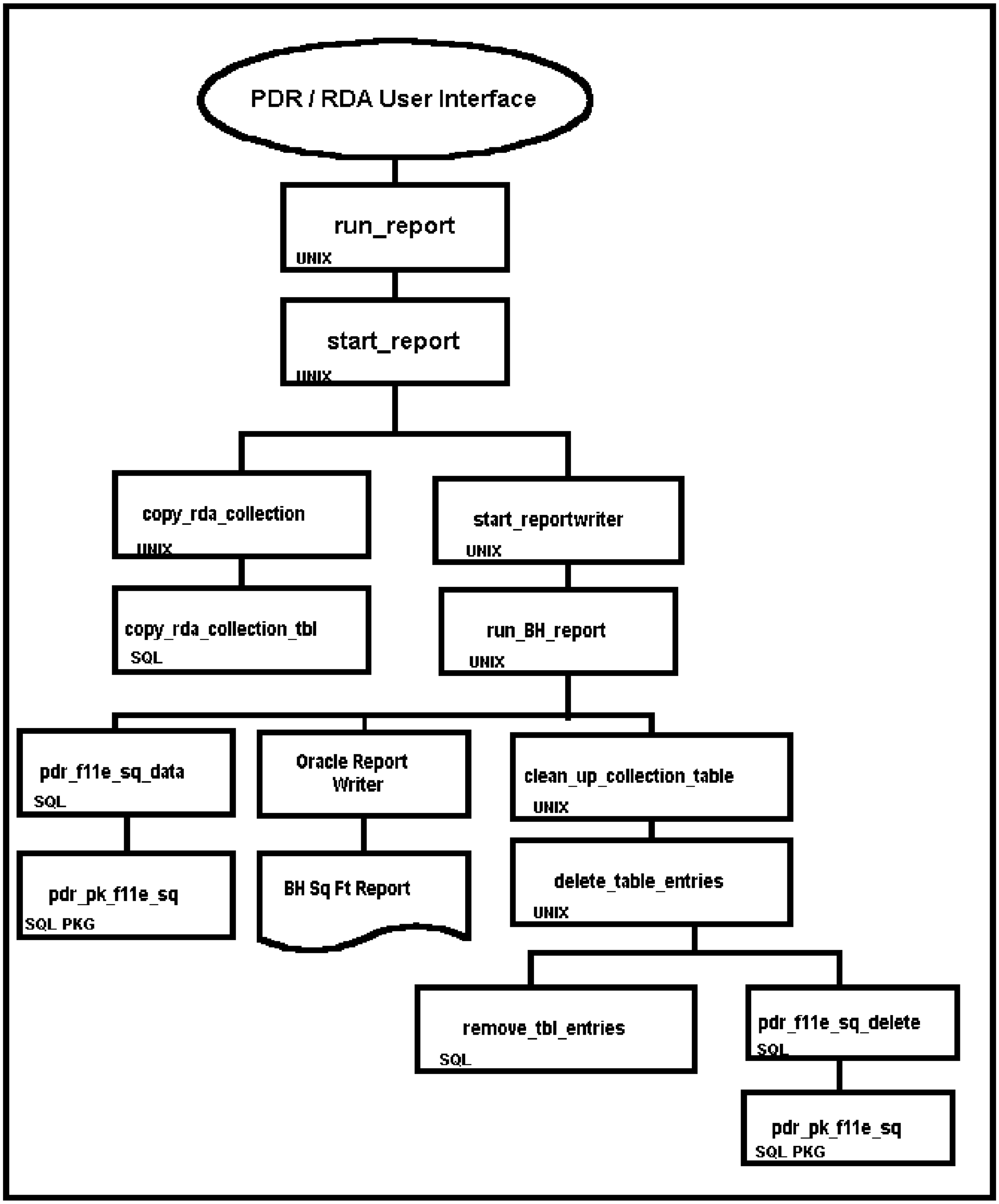

Figure A.2.2.2.1.1.2-8. Time-Phased Transportation Requirements Working Paper - Square Footage (F11E-SQ) Script Execution Hierarchy

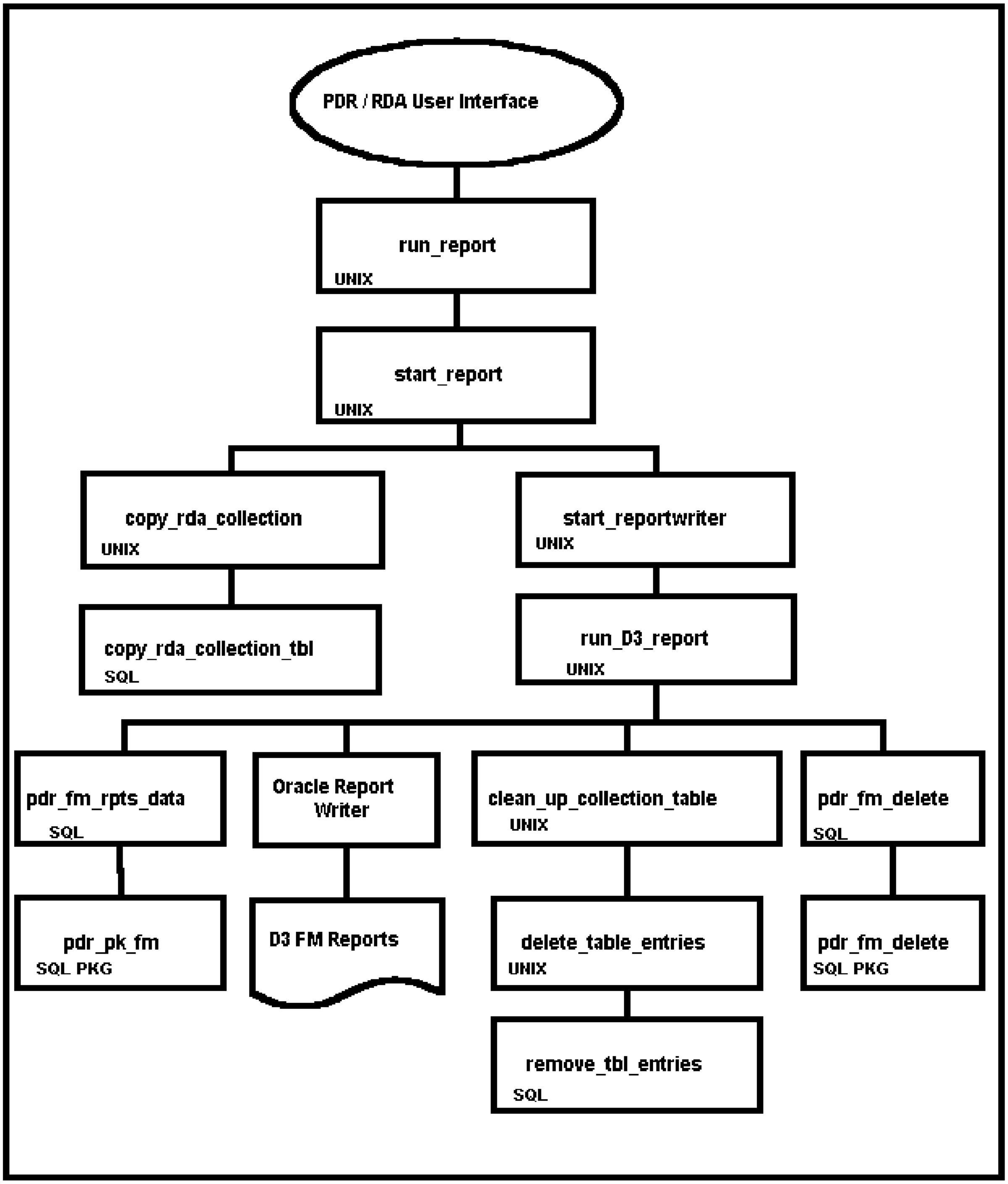

Figure A.2.2.2.1.1.2-9. FM Reports Script Execution Hierarchy

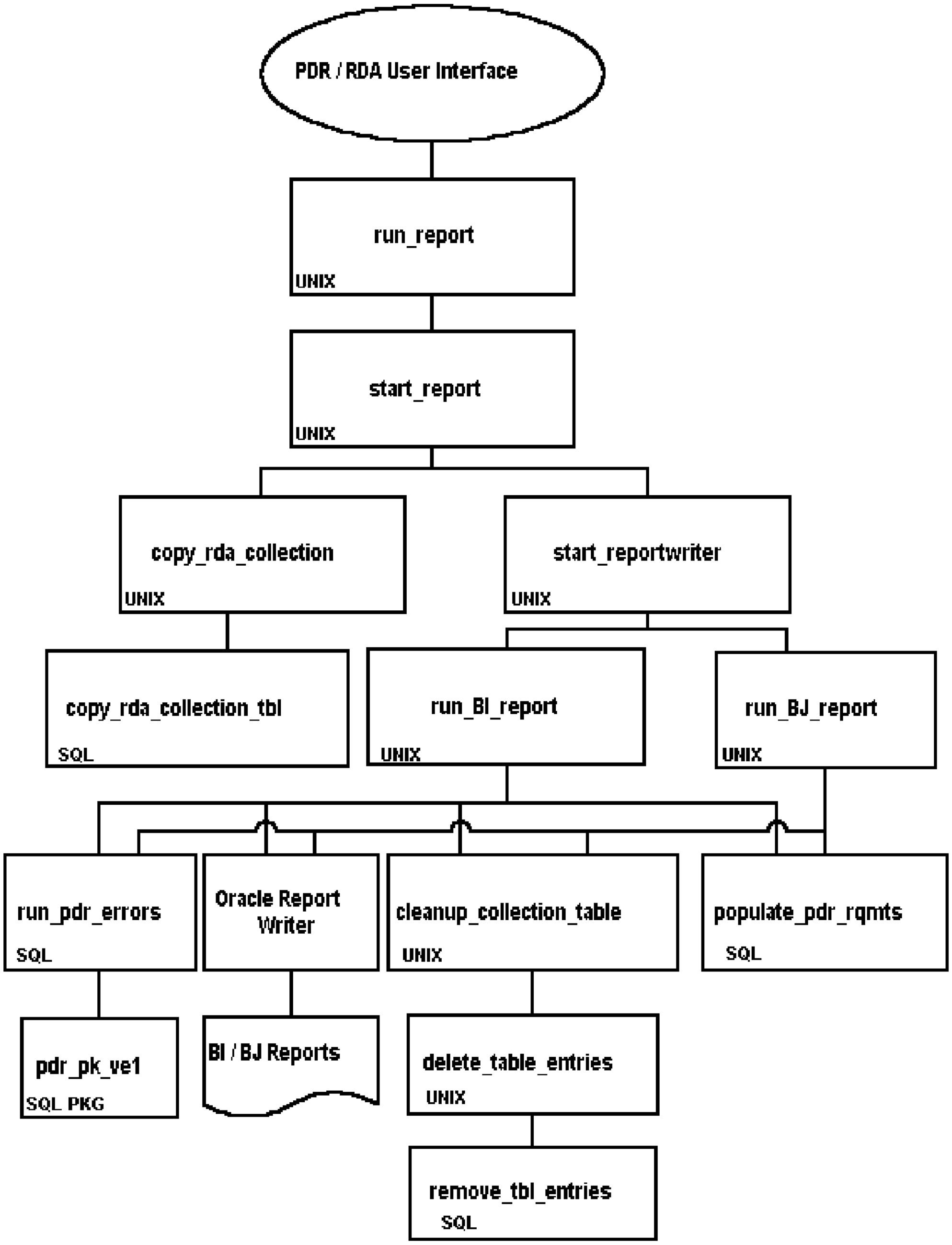

Figure A.2.2.2.1.1.2-10. Error Reports Script Execution Hierarchy

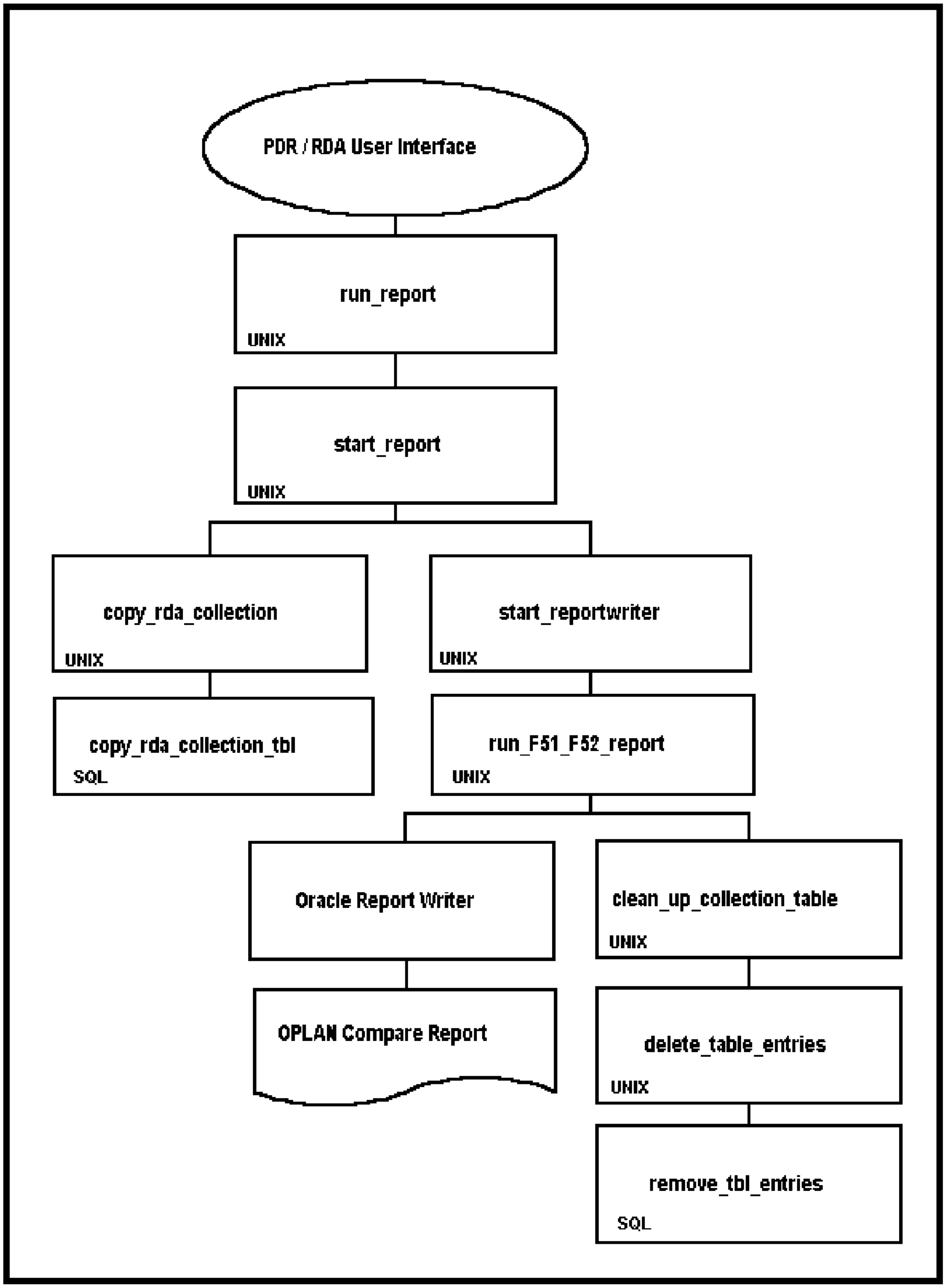

Figure A.2.2.2.1.1.2-11. OPLAN (TPFDD) Comparison Report Script Execution Hierarchy

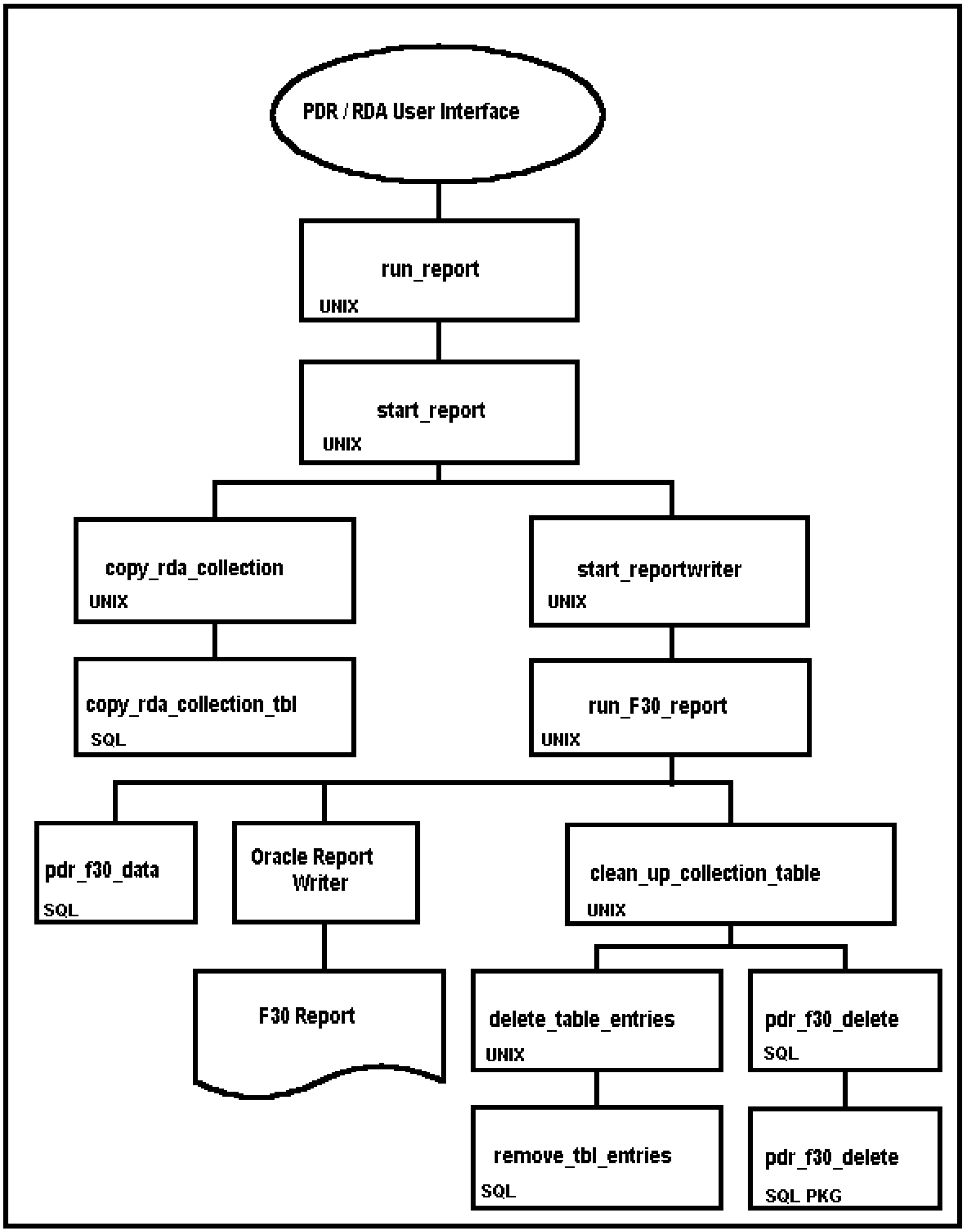

Figure A.2.2.2.1.1.2-12. F30 Report Script Execution Hierarchy

A.2.2.2.1.1.3 Report Generation

The functional unit items in this category consist of report definition and generation files (with *.rdf* or *.rep* extensions). Report definition files with the *.rdf* extension are created by and modifiable with the ORACLE Reports Designer tool and contain source statements. The report generation files with the *.rep* extension are generated from the *.rdf* files; they are not in themselves executable but are binary input files to the ORACLE Reports engine. Both the *.rdf* and the *.rep* files are delivered with the PDR application. The report definition files are located in the */h/PDR/data* directory.

For each report definition file, there are two binary report generation files, one for Sun platforms, which is located in the */h/PDR/data/sol2s* directory, and one for Hewlett-Packard platforms, which is located in the */h/PDR/data/hp700* directory. The following list shows the report generation files developed to support the OPLAN-based reports. The list shows only the *.rdf* files.

File	Description
BG_AIR.rdf	Airlift Requirements Detail Report
BG_AMC.rdf	AMC Requirements Detail Report
BG_MSC.rdf	M.C. Requirements Detail Report
BG_SEA.rdf	Sealift Requirements Detail Report
BI_F50.rdf	OPLAN Logical Errors Report
BJ.rdf	Transportation Pre-Edit Report
F11D.rdf	Force List/Movement Requirements Working Paper
F11E_SQ.rdf	Time-Phased Transportation Rqmts Working Paper - Square Footage
F11E_TN.rdf	Time-Phased Transportation Rqmts Working Paper - Tonnage
F11W.rdf	Force Requirements Detail Report
F30.rdf	Transportation Summary Report
F52.rdf	OPLAN (TPFDD) Compare Report
FM_F11.rdf	FM Report
FM_RQT.rdf	FM Plan Reference Report
FM_RU.rdf	FM Roll Up Report

More detailed information on these reports is available online from the ORACLE Reports Designer, in the comments section of the Global Properties submenu of the Report Menu (Report/General Properties/Comments) for each report. See the PDR Users Manual, referenced in Section 2.0, item m., for information about the contents of the reports.

A.2.2.2.1.2 GEO Paging/Reports

The GEO Paging/Reports logical unit is composed of the functional units described in this paragraph.

A.2.2.2.1.2.1 User Interface

The GEO Select functional unit was developed to support the GEO Paging/Reports. This functional unit provides data selection and display capabilities and initiates the optional hardcopy report generation process.

A.2.2.2.1.2.2 Reports Interface

The interface between the PDR Gain application, where the user specifies and qualifies a report, and the generation of the report is accomplished using UNIX and SQL scripts. The UNIX scripts are located in the */h/PDR/Scripts* directory; and the SQL scripts are located in the */h/PDR/sql* directory and can be recognized by the .sql extension. In addition to the PDR Report Writer Interface scripts that are listed under PDR Common Services scripts in Paragraph A.2.2.2.1.5.3, PDR Report Writer Interface, the following scripts are specific to the generation of the FE report; they are shown in execution hierarchy form:

```
run_FF_FE_report
        clean_up_geo_tucha_collection_table
        delete_geo_tucha_table_entries
                remove_geo_tucha_tbl_entries.sql
```

Note: The last three scripts listed are common to both the GEO Paging/Reports and TUCHA Paging/Reports logical units. Figure A.2.2.2.1.2.2-1, GEO/TUCHA Reports Execution Hierarchy, shows the common script execution hierarchy for both the GEO and TUCHA reports.

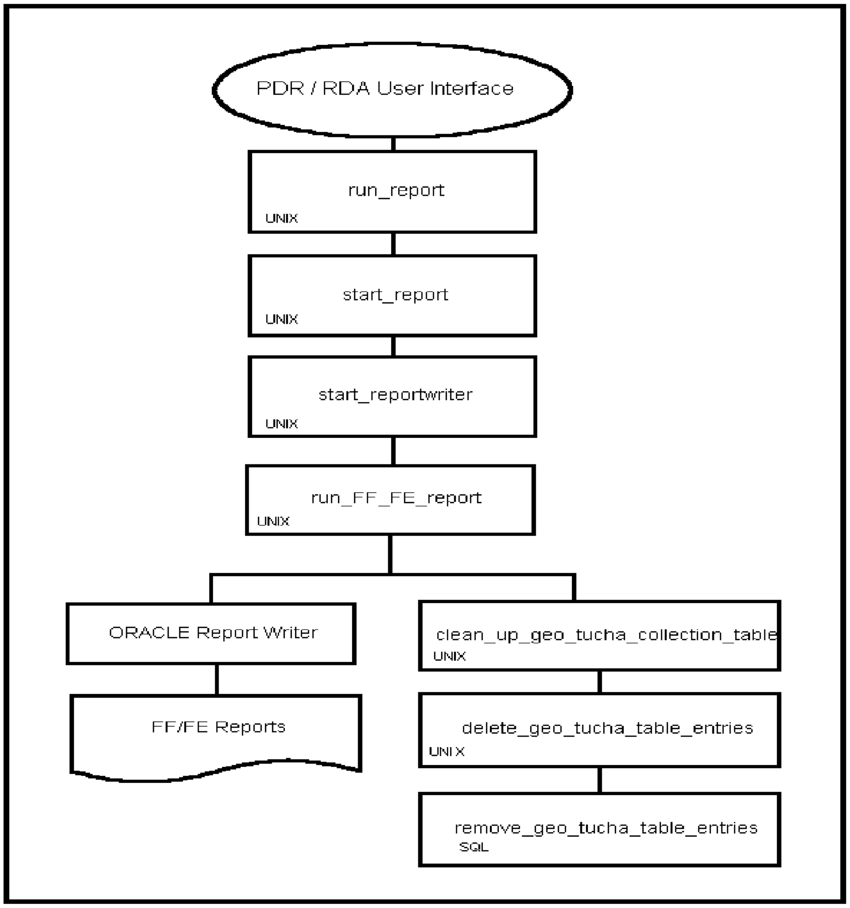

Figure A.2.2.2.1.2.2-1. GEO/TUCHA Reports Execution Hierarchy

A.2.2.2.1.2.3 Reference File Report Generation

The functional unit item in this category consists of a report definition file and accompanying binary report generation files. The report definition file with the *.rdf* extension is created by and modifiable with the ORACLE Reports Designer tool and contains source statements. The report generation files with the *.rep* extension are generated from the *.rdf* file; they are not in themselves executable but are binary input files to the ORACLE Reports engine. The report definition files are located in the */h/PDR/data* directory.

For each report definition file, there are two binary report generation files, one for Sun platforms, which is located in the */h/PDR/data/sol2s* directory, and one for Hewlett-Packard platforms, which is located in the */h/PDR/data/hp700* directory. The following list shows the report definition file developed to support the GEO Paging/Reports. The list shows only the *.rdf* file.

File	Description
FE.rdf	GEO Report

More detailed information on this report is available online from the ORACLE Reports Designer in the Report/General Properties/Comments section.

A.2.2.2.1.3 TUCHA Paging/Reports

The TUCHA Paging/Reports logical unit is composed of the functional units described in this paragraph.

A.2.2.2.1.3.1 User Interface

Functional units developed to support the TUCHA Paging/Reports are as follows:

a. Select (TUCHA) Functional Unit - The select functional unit provides TUCHA data selection and display capabilities and initiates the optional hardcopy report generation process.

b. TUCHA Functional Unit - The TUCHA functional unit enables user-specification of TUCHA data selection criteria by providing for multiple specifications of data subsets from functional categories by the use of filters on specified data fields, e.g., UTCs, movement status, service codes.

A.2.2.2.1.3.2 Reports Interface

The interface between the PDR Gain application, where the user specifies and qualifies a report, and the generation of the report is accomplished using UNIX and SQL scripts. The UNIX scripts are located in the */h/PDR/Scripts* directory; and the SQL scripts are located in the */h/PDR/sql* directory and can be recognized by the *.sql* extension. In addition to the PDR Report Writer Interface scripts that are listed under PDR Common Services scripts in Paragraph A.2.2.2.1.5.3, the following scripts are specific to the generation of the FF report; they are shown in execution hierarchy form:

```
run_FF_report
    clean_up_geo_tucha_collection_table
    delete_geo_tucha_table_entries
        remove_geo_tucha_tbl_entries.sql
```

Note: The last three scripts listed are common to both the GEO and TUCHA functional units. (See Figure A.2.2.2.1.2.2-1, which shows the common script execution hierarchy for both the GEO and TUCHA reports.)

A.2.2.2.1.3.3 TUCHA Report Generation

The functional unit item in this category consists of a report definition file and accompanying binary report generation files. The report definition file with the *.rdf* extension is created by and modifiable with the ORACLE Reports Designer tool and contains source statements. The report generation files with the *.rep* extension are generated from the *.rdf* file; they are not in themselves executable but are binary input files to the ORACLE Reports engine. The report definition files are located in the */h/PDR/data* directory. For each report definition file, there are two binary report generation files, one for Sun platforms, which is located in the */h/PDR/data/sol2s* directory, and one for Hewlett-Packard platforms, which is located in the */h/PDR/data/hp700* directory. The following list shows the report definition file developed to support the TUCHA Paging/Reports.

File	Description
FF.rdf	TUCHA Summary Report

More detailed information on this report is available online from the ORACLE Reports Designer, in the Report/General Properties/Comments section.

A.2.2.2.1.4 Reference File Status Report

The Reference File Status Report logical unit is composed of the functional units described in this paragraph.

A.2.2.2.1.4.1 User Interface

The Reference File Status Report is a selection on the PDR Menu bar. There are no selection options for this report. Once the user chooses the Reference File Summary Report, the “PDRMenu” application invokes the report display or hardcopy generation process.

A.2.2.2.1.4.2 Reports Interface

The interface between the PDR Gain application, where the user specifies and qualifies a report, and the generation of the report is accomplished using UNIX and SQL scripts. The UNIX scripts are located in the */h/PDR/Scripts* directory; and the SQL scripts are located in the */h/PDR/sql* directory and can be recognized by the *.sql* extension. In addition to the PDR Report Writer Interface scripts that are listed under PDR Common Services scripts in Paragraph A.2.2.2.1.5.3, PDR Report Writer Interface, the following scripts are specific to the generation of the F12A report; they are shown in execution hierarchy form:

```
run_F12A_report
    clean_up_collection_table
    delete_table_entries
        remove_tbl_entries.sql
```

See Figure A.2.2.2.1.4.2-1, F12A Report Script Execution Hierarchy, which shows the script execution hierarchy for the Reference File Status Report.

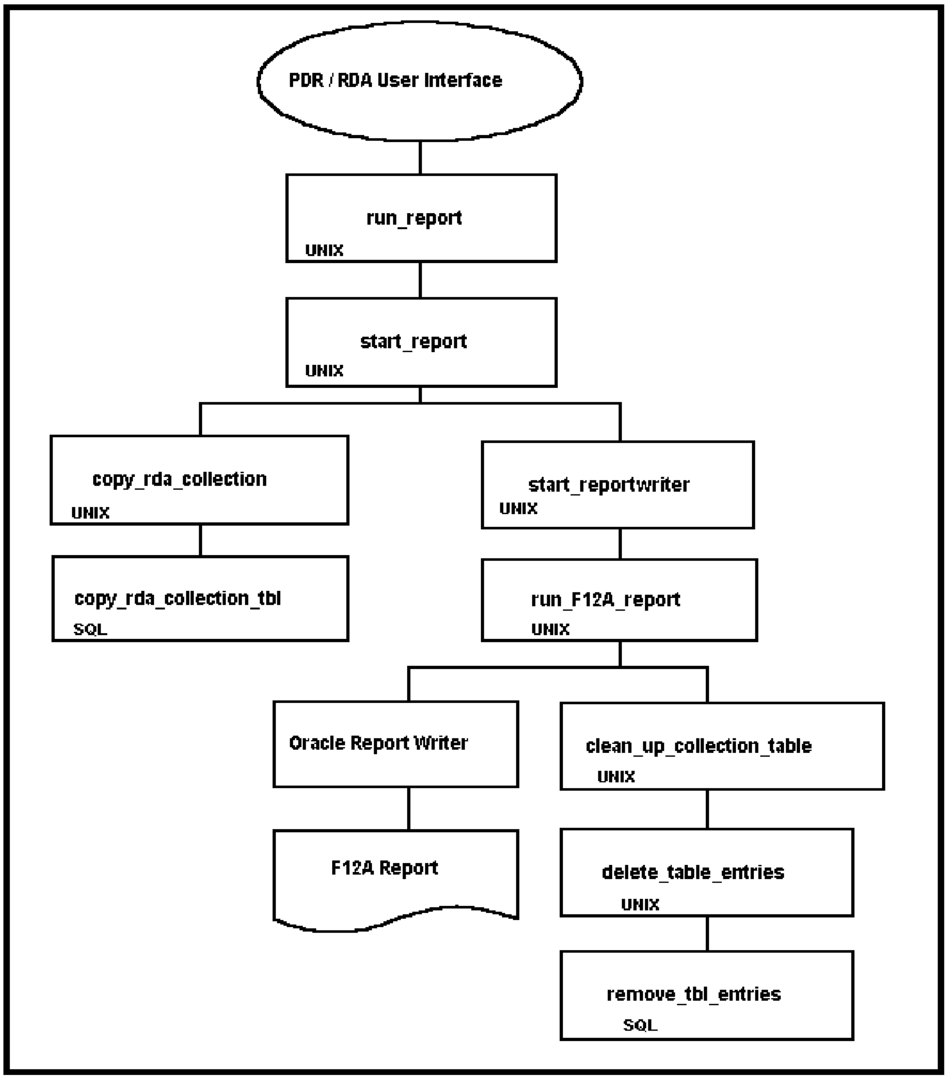

Figure A.2.2.2.1.4.2-1. F12A Report Script Execution Hierarchy

A.2.2.2.1.4.3 Reference File Summary Report Generation

The functional unit item in this category consists of a report definition file and accompanying binary report generation files. The report definition file with the *.rdf* extension is created by and modifiable with the ORACLE Reports Designer tool and contains source statements. The report generation files with the *.rep* extension are generated from the *.rdf* file; they are not in themselves executable but are binary input files to the ORACLE Reports engine. The report definition files are located in the */h/PDR/data* directory. For each report definition file, there are two binary report generation files, one for Sun platforms, which is located in the */h/PDR/data/sol2s* directory, and one for Hewlett-Packard platforms, which is located in the */h/PDR/data/hp700* directory. The following list shows the report definition file developed to support the Reference File Summary Report.

File	Description
F12A.rdf	Reference File Status Report

There are no data selection choices for the report, and it is generated directly upon its selection.

A.2.2.2.1.5 PDR Common Services

The PDR Common Services logical unit is composed of the functional units described in this paragraph. This includes all functional units developed to support the PDR work package in general and is not specific to any of the logical units already described.

A.2.2.2.1.5.1 PDR Startup

The functional units described in this paragraph support the invocation of the PDR application from the GCCS Desktop. The scripts are located in the */h/PDR/Scripts* directory. The scripts define a number of environment variables and verify that the number of Gain processes has not exceeded the site limit, and verify that the user has been granted the database permissions required for using PDR. The following scripts are shown in execution hierarchy form:

```
PDR_launch
        launch_PDR
                start_pdr
                        pdr_check_user_role
        check_active_gain_processes.csh
                check_active_gain_processes.tk
```

Note: The following file is the PDR logo image that was used when PDR was started from JNAV: */h/PDR/pdr_home/PDRdist/lib/splash.tif*. It is currently not used, but may be in the future.

Note: The following file is a 48 by 48 pixel bit map icon file for the PDR button: */h/PDR/Profiles/pdr.img*. This is not currently used, as PDR is available through JNAV. If PDR is ever to be invoked directly from the GCCS Desktop, this icon file will be used.

A.2.2.2.1.5.2 PDR High Level Navigation

High level navigation through the PDR user interface is accomplished by the “StartPDR” functional unit and the Report Selection menu of the “oplan_id” functional unit.

A.2.2.2.1.5.2.1 StartPDR Application

The “StartPDR” functional unit is entered from JNAV, and directs the user to either the OPLAN Selection, GEO Selection, or TUCHA Selection functional unit, depending on the Report Selection already made in JNAV.

A.2.2.2.1.5.2.2 PDR Menu

All PDR reports are available from a pull down menu labeled “Report Selection” on the main window of the “oplan_id” functional unit. Using this feature, the user may select successive reports without having to exit the PDR application.

A.2.2.2.1.5.3 PDR Report Writer Interface

The interface to the ORACLE Reports report generator starts with the PDR “RunReport,” “SelectSort,” and “PDRSelection” applications. These applications set the parameters for the report. The “RunReport” application invokes a series of UNIX scripts that perform the sequence of operations for report generation, including the initiation of the report writer, report cancellation, output destination, and report termination status notification for reports run in background mode.

The initiation and control of the report generation process is accomplished using UNIX and SQL scripts. The UNIX scripts are located in the */h/PDR/Scripts* directory; and the SQL scripts are located in the */h/PDR/sql* directory and can be recognized by the .sql extension. The following scripts are part of the execution path for all PDR reports; they are shown in execution hierarchy form:

```
run_report
    pdr_setup_printer
        get_current_printer.csh
            get_current_printer_wish.csh
            get_current_printer_wish.tk
            get_current_printer_xterm.csh
    start_report
        pdr_check_user_role
        copy_rda_collection
            change_time.sql
            copy_rda_collection_tbl.sql
            clean_up_collection_table
                delete_table_entries
                    remove_tbl_entries.sql
        start_bg_process.csh
            start_reportwriter
                replace
                    overwrite
            run_simple_report
    pdr_nohup
        report_monitor
            report_monitor_window
                kill_processes
```

A.2.2.2.1.5.3.1 User Interface Parameter Determination

The PDR “RunReport” application controls the invocation and execution of the UNIX scripts. “SelectSort” gets input from the user that specifies the sort order of the OPLAN-based and GEO reports. The “PDRRunReport” application get input from the user that specifies the security classification of the report and the destination of the report, e.g., printer or screen.

See Figure A.2.2.2.1.5.3-1, PDR Common Services Report Interface Script Execution Hierarchy, which shows the script execution hierarchy for the PDR Report Writer Interface.

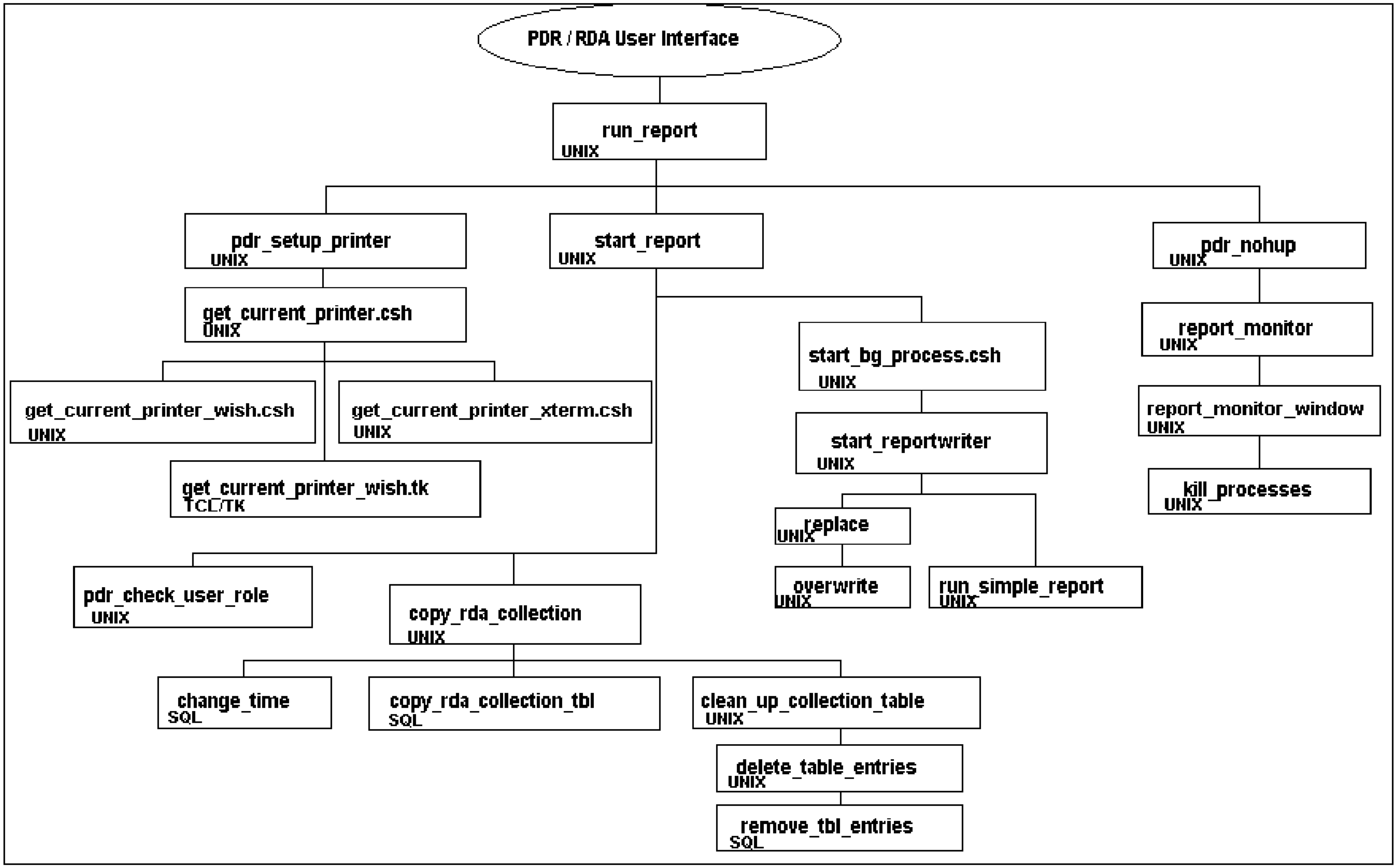

Figure A.2.2.2.1.5.3-1. PDR Common Services Report Interface Script Execution Hierarchy

A.2.2.2.1.5.3.2 Report Termination Status Notification

The text files shown below are used in reporting the termination status of a background report.

> */h/PDR/data/errors.txt* - Text file that is used to show an error message when a background invocation of ORACLE Reports returns an error status.
>
> */h/PDR/data/finished.txt* - Text file that shows the termination status of an invocation of ORACLE Reports.

A.2.2.2.1.6 PDR Segmentation Services

The PDR software is packaged as two separate segments, the PDR application (client) segment and the PDRSRV database (server) segment.

A.2.2.2.1.6.1 PDR Application Segment

This segment contains the application software. This includes the Gain Momentum executable with the PDR runtime libraries; the PDR report definitions; and the UNIX and SQL support scripts.

A.2.2.2.1.6.1.1 PDR Application Segment Installation

The installation of the PDR application segment is accomplished using the following scripts. These are shown in the form of an execution hierarchy.

```
/h/PDR/SegDescrip/PostInstall
        /h/PDR/Scripts/set_printer_flag
        /h/PDR/Scripts/set_max_gain_processes.csh
```

A.2.2.2.1.6.1.2 PDR Application Segment Deinstallation

The deinstallation of the PDR application segment is accomplished using the following script:

> */h/PDR/SegDescrip/DEINSTALL*

A.2.2.2.1.6.2 PDRSRV Database Segment

This segment contains the PDR-specific database objects required to support the PDR product and the scripts to perform the PDRSRV segment install/deinstall and enable/disable user procedures.

A.2.2.2.1.6.2.1 PDRSRV Database Segment Installation

The installation of the PDRSRV database segment is accomplished using the scripts listed below. Full paths are shown for each script because the UNIX scripts reside in two different directories. These are shown in the form of an execution hierarchy.

- */h/PDRSRV/SegDescrip/PostInstall*
    - */h/PDRSRV/sql/pdr_verify_tablespace.sql*
    - */h/PDRSRV/Scripts/PostInstall.set_dir*
    - */h/PDRSRV/sql/pdr_cr_tablespace.sql*
    - */h/PDRSRV/sql/pdr_cr_role.sql*
    - */h/PDRSRV/Scripts/enter_new_oracle_password*
    - */h/PDRSRV/Scripts/determine_oracle_password*
    - */h/PDRSRV/sql/pdr_cr_jr_user.sql*
    - */h/PDRSRV/sql/pdr_grant_tm.sql*
    - */h/PDRSRV/sql/pdr_cr_tables.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_sys.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_coll.sql*
        - */h/PDRSRV/sql/pdr_cr_table_report_options.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_selection.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_sort.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_t28.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_bg.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_f11e_tn.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_f11e_sq.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_f11w.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_f30.sql*
        - */h/PDRSRV/sql/pdr_cr_tables_fm.sql*
    - */h/PDRSRV/sql/pdr_populate_tables.sql*
        - */h/PDRSRV/sql/pdr_insert_sort_columns.sql*
        - */h/PDRSRV/sql/pdr_insert_sort_criteria.sql*
        - */h/PDRSRV/sql/pdr_insert_logical_errors.sql*
    - */h/PDRSRV/sql/pdr_cr_views.sql*
        - */h/PDRSRV/sql/pdr_cr_view_f11e_sq.sql*
        - */h/PDRSRV/sql/pdr_cr_view_f11e_tn.sql*
    - */h/PDRSRV/sql/pdr_grant_jr.sql*
    - */h/PDRSRV/sql/pdr_cr_packages_tm.sql*
        - */h/PDRSRV/sql/pdr_pk_ve1_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_ve1_body.sql*
        - */h/PDRSRV/sql/rda_pk_ucp_counts_spec.sql*
        - */h/PDRSRV/sql/rda_pk_ucp_counts_body.sql*
    - */h/PDRSRV/sql/pdr_cr_packages_jr.sql*
        - */h/PDRSRV/sql/pdr_pk_syn_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_syn_body.sql*
        - */h/PDRSRV/sql/pdr_pk_bg_air_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_bg_air_body.sql*
        - */h/PDRSRV/sql/pdr_pk_bg_amc_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_bg_amc_body.sql*
        - */h/PDRSRV/sql/pdr_pk_f11e_sq_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_f11e_sq_body.sql*
        - */h/PDRSRV/sql/pdr_pk_f11e_tn_spec.sql*
        - */h/PDRSRV/sql/pdr_pk_f11e_tn_body.sql*
        - */h/PDRSRV/sql/pdr_pk_f11w_spec.sql*

```
            /h/PDRSRV/sql/pdr_pk_f11w_body.sql
            /h/PDRSRV/sql/pdr_pk_f30_spec.sql
            /h/PDRSRV/sql/pdr_pk_f30_body.sql
            /h/PDRSRV/sql/pdr_pk_fm1_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm1_body.sql
            /h/PDRSRV/sql/pdr_pk_fm2_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm2_body.sql
            /h/PDRSRV/sql/pdr_pk_fm3_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm3_body.sql
            /h/PDRSRV/sql/pdr_pk_fm4_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm4_body.sql
            /h/PDRSRV/sql/pdr_pk_fm5_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm5_body.sql
            /h/PDRSRV/sql/pdr_pk_fm_spec.sql
            /h/PDRSRV/sql/pdr_pk_fm_body.sql
        /h/PDRSRV/sql/pdr_get_user_info.sql
        /h/PDRSRV/sql/pdr_update_syn.sql
        /h/PDRSRV/sql/pdr_run_updates.sql
```

A.2.2.2.1.6.2.2 PDRSRV Database Segment Deinstallation

The deinstallation of the PDRSRV database segment is accomplished using the scripts listed below. These are shown in the form of an execution hierarchy.

```
    /h/PDRSRV/SegDescrip/DEINSTALL
            /h/PDRSRV/Scripts/pdr_full_deinstall
                /h/PDRSRV/Scripts/DEINSTALL.warning
                /h/PDRSRV/sql/pdr_get_user_info.sql
                /h/PDRSRV/sql/pdr_drop_all_syn.sql
                /h/PDRSRV/sql/pdr_run_updates.sql
                /h/PDRSRV/sql/pdr_full_deinstall.sql
```

A.2.2.2.1.6.2.3 PDR User Access Control

The components described in this paragraph provide PDR support services to enable and disable a user for the PDR application.

A.2.2.2.1.6.2.3.1 Enable User for the PDR Subsystem

The following scripts, shown as an execution hierarchy, are used to enable a user for the PDR application.

```
    /h/PDRSRV/install/pdr_enable_user.csh
        /h/PDRSRV/install/pdr_check_user.sql
        /h/PDRSRV/install/pdr_assign_to_role.sql
        /h/PDRSRV/sql/pdr_get_user_info.sql
        /h/PDRSRV/install/pdr_cr_synonyms.sql
        /h/PDRSRV/sql/pdr_run_updates.sql
```

A.2.2.2.1.6.2.3.2 Disable User from PDR

The following scripts, shown as an execution hierarchy, are used to disable a user from the PDR application.

```
/h/PDRSRV/install/pdr_disable_user.csh
        /h/PDRSRV/install/pdr_check_user.sql
        /h/PDRSRV/install/pdr_revoke_user.sql
        /h/PDRSRV/sql/pdr_get_user_info.sql
        /h/PDRSRV/install/pdr_drop_synonyms.sql
        /h/PDRSRV/sql/pdr_run_updates.sql
```

A.2.2.2.1.6.2.3.3 Common Scripts

The following scripts are used in both the enable user and the disable user functions.

/h/PDRSRV/install/pdr_check_user.sql - Verifies that user has an ORACLE account.
/h/PDRSRV/sql/pdr_get_user_info.sql - Builds table entries of database synonyms for authorized PDR users.
/h/PDRSRV/sql/pdr_run_updates.sql - Executes database-stored queries to add/drop the PDR user.

A.2.3 Software Component Interaction

This paragraph describes interactions of the PDR software components. The first paragraph describes the PDR concept of operations. The second paragraph describes the high level operational interaction of the PDR logical units. The third paragraph provides a detailed description of the logical units' interoperability.

A.2.3.1 Concept of Operations

The PDR application consists of a user interface, which provides the means for the user to communicate all reporting requirements to the PDR application, and a back-end process that controls the generation of report output. The PDR user interface is implemented using Gain Momentum 3.1. The reports are implemented using ORACLE Reports 2.0. Communication between the different environments is accomplished using a combination of UNIX and SQL scripts with appropriate environment variables. In addition, a number of local database objects were created to support the PDR application. A PDR application schema, separate from the JOPES Core database, is created at PDR installation time. Figure A.2.3.1-1, PDR Concept of Operation, shows the PDR concept of operations.

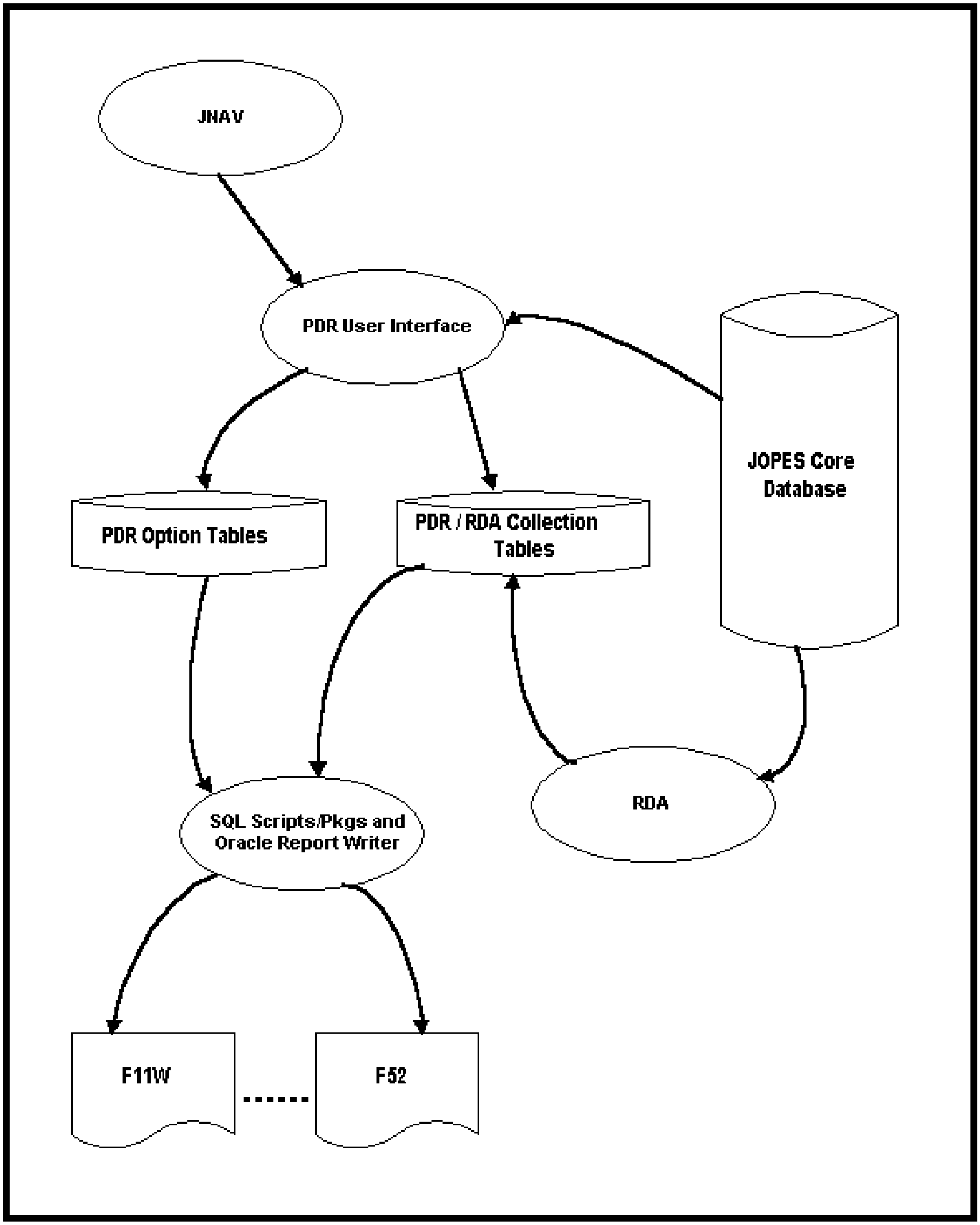
JNAV
PDR User Interface
JOPES Core Database
PDR Option Tables
PDR / RDA Collection Tables
SQL Scripts/Pkgs and Oracle Report Writer
RDA
F11W
F52

Figure A.2.3.1-1. PDR Concept of Operation

The user interface includes the capability to select a report, define the target records for reporting, define the required sort, and define any other processing options required for the particular report selected. The OPLAN-based reports can also be initiated from RDA. In this option, the RDA user interface defines the target records for reporting, and invokes the PDR "RunReport" application to start the generation of the report.

After the user has selected the criteria to be used to filter the data for the report, identifying information for the defined record set is written to collection tables in the ORACLE database. The identifying information for OPLAN-based reports consists of OPLAN Number, Requirement Type (Force, Cargo, Personnel), Requirement ID [Unit Line Number (ULN)/Cargo Increment Number (CIN)/Personnel Increment Number (PIN)], and the ORACLE ROWID of the record on the JOPES Core database. Identifying information consists of the Geographic Location Code (GEOLOC) for GEO reporting and the Unit Type Code (UTC) for the TUCHA. The exception to this process is the Reference File Status Report, which is inclusive in nature and for which there are no user-specified criteria.

After the PDR Gain Momentum application "RunReport" gets user input on the classification of the report and the report destination, the UNIX scripts to run the report as a subordinate process are started. After the report generation is completed, and the report has been printed, or written to a file, or the user is finished with the screen display of the data, the records in the collection tables for this report are deleted.

A.2.3.2 Logical Units Interoperability

This paragraph describes some features of the PDR implementation in greater detail.

A.2.3.2.1 Collection and Options Tables

There are currently three PDR collection tables to identify OPLAN-based records, GEO records, and TUCHA records. The PDR application uses collection tables to identify the records selected by the user for inclusion in a formatted report. The collection tables contain only record-identifying information such as requirement identifier, GEOLOC code, or UTC. No potentially sensitive OPLAN information is included in them. To support multiple concurrent users and/or overlapping report generation by a single user, the collection table records contain the user name of the PDR user's GCCS account and a date/time stamp that identifies when the user requested the generation of a specific report on the specific set of data selected. Each request for a report is uniquely identified by the combination of the "user_id" and the date/time stamp of the request. See Paragraph A.2.5.2.3.1, Collection Tables.

Also, there are currently two PDR options tables to capture the data selection and other qualifications that the user chose to define the contents of the requested report. These tables contain the same report-specific keys as the PDR collection tables and data that identifies the user's choices. See Paragraph A.2.5.2.3.2, Options Tables.

When the PDR application is invoked from an RDA session, RDA collection tables play a part in the PDR process. The data selection capability of the user interface to OPLAN-based reports is implemented by the RDASelect application. After the RDA user has selected a set of records, the RDA user can specify a report of the records. When this request is received by PDR, the PDR application uses the marked records in the RDA collection tables as the data source for the report. PDR copies the marked RDA records to the PDR collection tables and maintains the data in the PDR collection tables until the reporting process is complete.

A.2.3.2.2 Sort

In order to allow users to sort on any primary field included in a report, a table exists in the PDR schema that defines each report's primary fields. The table associates the sequence number of the field in the report's SQL select statement with each field by name. As the user selects the fields for the desired sorting of the report, the sequence numbers of these fields in the report's select statement are used to create a text string of the SQL "ORDER BY" clause, which is passed as an argument to the report generator. Because the sort columns table also includes the reports' predefined sort sequences, this technique is used to pass all sort conditions to the report generator.

The use of sequence numbers to reference fields for sorting means there is a precise matching between the sorting mechanism and the exact sequence of fields in the "select" part of a report query. Any changes to the query of a report must look at and adjust, if necessary, the entries in the PDR sort table to maintain the correct linkage between fields and the sort selections specified by the user. Additionally, this technique puts a constraint on the construction of a report generator. The columns to be sorted must be coded in a single query because multiple queries render it impossible to associate each data field with a single column reference number.

A.2.3.2.3 ORACLE Reports

Many of the arguments defined in the PDR user interface, passed to the UNIX scripts and used to define specifications for an individual report request, exist because of the multiple possible combinations of system settings and user choices that have to be accommodated in the PDR application. The reports are generated in the UNIX background for two reasons: 1) use of the background for report generation returns control of the interface application to the user and 2) it allows the user to initiate the generation of a report and log off the GCCS. Thus, it maximizes the users time, eliminating waiting for the completion of the reporting process.

The ORACLE Reports runtime formats data using information described in a report definition file. The resulting formatted report data can be sent to a window, a file, or a printer. There are two modes for displaying data in a Screen and Preview window. The printer must be defined so that the Preview mode can format the data on the window exactly as it would be printed. The Screen mode does not impose this restriction but, conversely, the report cannot be printed from a report window in Screen mode.

The other two output destinations, File and Printer, also require that an ORACLE printer be defined. To put the report output into an ASCII file, click on the File main menu, then click **{Choose Printer}**. In the "Pre-defined Printers" box, select the ASCII printer. Exit the Choose Printer dialog box. Click on the File main menu, then click on **{Run}**. In the Runtime Parameter box, Click **{RunReport}**. In the Previewer box, click **{Print}**; then, in the PrintJob box, click **{Save}**, then, in the Save File Dialog box, enter filename, then **{OK}** to exit the dialog box. Then click **{Close}** to close the Previewer. This procedure is for reports that have parameter forms defined.

Since printers may not be installed at every site, an environment variable controls whether or not the user can select the preview, printer, or file modes. This variable is "PDR_PRINTER_DEFINED" and it can be changed by running the script, *h/PDR/Scripts/set_printer_flag*.

A.2.4 Environment Variables

Communication between the different software elements in the PDR application is simplified by the use of environment variables. These environment variables are described in the following paragraphs, organized by the sponsoring software element.

A.2.4.1 ORACLE Kernel Environment Variables

The following ORACLE environment variables must be set for the PDR application to execute successfully. Refer to ORACLE documentation for more information on these variables.

a. ORACLE_HOME,
b. ORACLE_SID,
c. ORACLE_TERM,
d. TNS_ADMIN, and
e. TWO_TASK.

A.2.4.2 Gain Momentum Environment Variables

The following Gain Momentum environment variables must be set for the PDR application to execute successfully. Refer to Gain Momentum documentation for more information on these variables.

a. GAINHOME,
b. DBVENDOR,
c. DBTIMEOUT,
d. DBUSER,
e. DBPASSWORD,
f. DBSERVER,
g. DBENTRY,
h. DBSERVERNAME,
i. DBENVLIST,
j. DBPROXY,
k. VTXDIR,
l. DBSTARTPORT, and
m. DBPORTS.

A.2.4.3 ORACLE Reports Environment Variables

The following describes ORACLE environment variables that must be set for the PDR application to generate reports successfully. Refer to ORACLE Reports documentation for more information on these variables.

a. LD_LIBRARY_PATH, and
b. REP_PATH.

A.2.4.4 RDA Environment Variables

The following RDA environment variables must be set for the PDR application to execute successfully. Refer to RDA documentation for more information on these variables.

a. RDA_HOME**,**
b. RDAHELPENGINE,
c. RDAHELPDATA,
d. RDA_PRINT_P1,
e. RDA_PRINT_P2, and
f. RDAMODE.

A.2.4.5 PDR Environment Variables

The following PDR environment variables must be set for the PDR application to execute successfully:

a. PDR_HOME - Establishes the base path for PDR objects.
b. PDR_ORA_ENV - Path name to GCCS ORACLE environment sourcing file.
c. PDR_DEBUG_FLAG - Toggles debug mode; "ON" or "OFF".
d. PDR_OP_PLN_ID - Contains the value of the user selected OPLAN ID.
e. PDR_OPS_LOGIN_DISABLED - Specifies login account type.
f. PDR_ORIGINATOR - Identifies the invoking system, "PDR" or "RDA."
g. PDR_PRINTER_DEFINED - Specifies printer availability; "YES" or "NO".
h. PDR_REPORT_TYPE - Defines the current user report selection.
i. GCP_ROOT_PID - The UNIX Process ID of the currently-selected printer.
j. PDR_USER - UNIX account name of the PDR user who initiated the report.

A.2.5 Database Variables

This paragraph describes the database objects that the PDR application uses.

A.2.5.1 JOPES Core Database

The PDR application grants access to JOPES Core database objects through the RDAUSER database role. Enter and execute the following SQLPLUS statement to list the privileges granted to the RDAUSER role:

```
select * from sys.role_tab_privs where role = 'RDAUSER';
```

A.2.5.2 PDR Database

The PDR application uses a number of database objects specific to the report generation process within its own tablespace: a schema, tables, views, a package, and a role.

A.2.5.2.1 Tablespace

All PDR database objects are created in the PDR tablespace. This tablespace is created during the PDRSRV segment installation. Use the browser to review the following file:

/h/PDRSRV/sql/pdr_cr_tablespace.sql

A.2.5.2.2 Schema

The PDR database objects are owned by the JOPES_REPORTS ORACLE account. This account is created during the PDRSRV segment installation. Use the browser to review the following file:

/h/PDRSRV/sql/pdr_cr_jr_user.sql

A.2.5.2.3 Tables

PDR support tables have been created under the JOPES_REPORTS ORACLE account. These tables are categorized as collection tables, options tables, intermediate data stores, lookup tables, and administrative or system tables.

A.2.5.2.3.1 Collection Tables

Collection tables are used by the PDR application to identify the set of records targeted for reporting. Three collection tables, listed below, are used by PDR for OPLAN-based reports, for GEO paging/reports, and for TUCHA paging/reports.

a. pdr_rqmts_collection,
b. pdr_geo_collection, and
c. pdr_tucha_collection.

Additionally, a fourth table, which is part of the table_master schema, is used for the FM reports. This table is named pdr_fm_collection. These tables include session identifying information (consisting of user name and date/time stamps to uniquely identify report requests) and information that identifies the data for each report. This identifying information consists of OPLAN Number and Requirement ID for OPLAN-based reports, GEO codes for GEO paging/reports, and UTC for TUCHA paging/reports. The definitions for the PDR schema tables can be obtained by browsing the following file:

/h/PDRSRV/sql/pdr_cr_tables_coll.sql

A.2.5.2.3.2 Options Tables

The PDR application uses several local tables, listed below, to hold the user’s selected options. This information is stored for the duration of a user report generation and is used to communicate the options selected in the user interface environment to the report generation environment for selection and ordering data. These tables currently exist for the user’s selection criteria, report options, and sort criteria.

a. pdr_selection_criteria,
b. pdr_report_options, and

c. pdr_sort_criteria.

The definitions for these tables can be obtained by browsing the following files:

/h/PDRSRV/sql/pdr_cr_tables_selection.sql
/h/PDRSRV/sql/pdr_cr_table_report_options.sql
/h/PDRSRV/sql/pdr_cr_tables_sort.sql

A.2.5.2.3.3 Intermediate Data Stores

The PDR application uses ORACLE tables to store intermediate information for specific reports in order to diminish the time required to generate the reports. This intermediate information is also associated with a user name and a logon date/time to support multiple users concurrently. The information for any given instance of user name and date/time of the report request is preserved only for the generation of the report. The definition for these tables can be obtained by browsing the following files:

/h/PDRSRV//sql/pdr_cr_tables_bg.sql
/h/PDRSRV//sql/pdr_cr_tables_f11e_tn.sql
/h/PDRSRV//sql/pdr_cr_tables_f11e_sq.sql
/h/PDRSRV//sql/pdr_cr_tables_f11w.sql
/h/PDRSRV//sql/pdr_cr_tables_f30.sql
/h/PDRSRV//sql/pdr_cr_tables_fm.sql

A.2.5.2.3.4 Lookup Tables

The PDR application uses local tables, listed below, to provide additional support data. The first is used by the BI/BJ reports; the second is used by the reports that support used-specified sorts.

a. pdr_error_corrective_action, and
b. pdr_sort_columns.

The definitions for these tables can be obtained by browsing the following file:

/h/PDRSRV/sql/pdr_cr_tables_t28.sql
/h/PDRSRV/sql/pdr_cr_sort.sql

A.2.5.2.3.5 PDR System Tables

The PDR application uses several tables, listed below, to provide administrative support to the control of PDR users.

a. pdr_sys_users,
b. pdr_error_corrective_action, and
c. pdr_sys_text.

The definitions for these tables can be obtained by browsing the following files:

/h/PDRSRV/sql/pdr_cr_tables_sys.sql

A.2.5.2.4 Views

Views were created for the generation of the F11E reports. The SQL script file names are self documenting. The view definitions can be obtained by browsing the following files:

/h/PDRSRV/sql/pdr_cr_view_f11e_sq.sql
/h/PDRSRV/sql/pdr_cr_view_f11e_tn.sql

A.2.5.2.5 Packages

To achieve better performance, the generation process for the BG, F11E, F11W, F30, and FM reports makes use of database-resident packages of PL/SQL queries. Also, the Verification Engine in support of OPLAN Error Analysis (BI/BJ) reports is implemented as a database package. The stored procedures in the packages are executed after the user has initiated a report and prior to the execution of the ORACLE Reports report generator for these reports. Because the stored procedures are compiled when they are loaded into the database, their installation is part of the server installation process. The following list shows the PL/SQL packages that PDR uses. Note that a stored procedure is composed of both a specification script that defines the input and output parameters and a script, which ORACLE refers to as the body, that contains the queries to be executed.

/h/PDRSRV/sql/pdr_pk_bg_air_body.sql
/h/PDRSRV/sql/pdr_pk_bg_air_spec.sql
/h/PDRSRV/sql/pdr_pk_bg_amc_body.sql
/h/PDRSRV/sql/pdr_pk_bg_amc_spec.sql
/h/PDRSRV/sql/pdr_pk_f11e_sq_body.sql
/h/PDRSRV/sql/pdr_pk_f11e_sq_spec.sql
/h/PDRSRV/sql/pdr_pk_f11e_tn_body.sql
/h/PDRSRV/sql/pdr_pk_f11e_tn_spec.sql
/h/PDRSRV/sql/pdr_pk_f11w_body.sql
/h/PDRSRV/sql/pdr_pk_f11w_spec.sql
/h/PDRSRV/sql/pdr_pk_f30_body.sql
/h/PDRSRV/sql/pdr_pk_f30_spec.sql
/h/PDRSRV/sql/pdr_pk_fm1_body.sql
/h/PDRSRV/sql/pdr_pk_fm1_spec.sql
/h/PDRSRV/sql/pdr_pk_fm2_body.sql
/h/PDRSRV/sql/pdr_pk_fm2_spec.sql
/h/PDRSRV/sql/pdr_pk_fm3_body.sql
/h/PDRSRV/sql/pdr_pk_fm3_spec.sql
/h/PDRSRV/sql/pdr_pk_fm4_body.sql
/h/PDRSRV/sql/pdr_pk_fm4_spec.sql
/h/PDRSRV/sql/pdr_pk_fm5_body.sql
/h/PDRSRV/sql/pdr_pk_fm5_spec.sql
/h/PDRSRV/sql/pdr_pk_fm_body.sql
/h/PDRSRV/sql/pdr_pk_fm_spec.sql
/h/PDRSRV/sql/pdr_pk_syn_body.sql
/h/PDRSRV/sql/pdr_pk_syn_spec.sql
/h/PDRSRV/sql/pdr_pk_ve1_body.sql
/h/PDRSRV/sql/pdr_pk_ve1_spec.sql

A.2.5.2.6 Roles

Access to required database objects is managed using a database role. The PRE_DEFINED_REPORTS_USER role is defined in the following file:

/h/PDRSRV/sql/pdr_cr_role.sql

A PDR user is associated with the role when the “pdr_enable_user.csh” script is run.

A.2.5.2.7 Synonyms

JOPES_REPORTS database objects are referenced through private synonyms. Synonyms are created for a user when the “pdr_enable_user.csh” script is run. For the sequence of scripts see Paragraph A.2.2.2.1.6.2.3.1, Enable User for the PDR Subsystem. In order to see the synonyms that exist for the database, enter the following SQLPLUS command:

```
select * from jopes_reports.pdr_sys_text;
```

A.2.6 External System Interfaces

A.2.6.1 JNAV Interface

The PDR application is launched from JNAV when the PDR icon is depressed and a report selected. JNAV executes the PDR launch script, which invokes several scripts necessary to execute the PDR Gain runtime in the GCCS environment. These scripts set up environment variables, check that the valid number of Gain processes has not been exceeded, and verify that the user has an ORACLE account and PDR access privileges. The PDR launch process can be traced by viewing the following files, which are shown in script execution hierarchy fashion:

```
/h/PDR/Scripts/PDR_launch
        /h/PDR/Scripts/launch_PDR
                /h/PDR/Scripts/check_active_gain_processes.sch
                /h/PDR/Scripts/check_active_gain_processes.tk
        /h/PDR/Scripts/start_pdr
                /h/PDR/Scripts/pdr_check_user_role
```

When the PDR application is started from JNAV, JNAV passes a text string argument that specifies the user-requested report type. The application script for the “PDRMenu” application receives this argument in the “reporttype” parameter. In the case of a null report type, the application script for “PDRMenu” sets the report type variable to “PDR” and the OPLAN screen is displayed.

A.2.6.2 RDA Interface

The OPLAN-based reports can be invoked from RDA, in addition to the JNAV/PDR route. RDA users have the ability to select the following PDR OPLAN-based reports: F11D, F11E (tonnage and square footage versions), F30, F11W, BI, BJ, BG (Airlift, AMC, Sealift, and MSC versions), and FF. RDA invokes the “startReport” function of the PDR “RunReport” application. The interface consists of the following parameters:

theOriginator,
theReportType, and
theOtherParameters.

The parameter “theOtherParameters” is composed of the user_id, the date-time stamp of the RDA collection of records, the process identifier (PID), and a text string that indicates whether a subset of a TPFDD has been marked. PDR does not return any values or statuses to RDA.

A.2.6.3 ORACLE Reports Interface

On initiation, each report is passed a number of arguments. Some of these are required by ORACLE Reports, e.g., page orientation, and are present for all reports. Some are required by GCCS, e.g., report classification, and are present for all reports. Some are unique to a specific report. Use the browser to review the following report initiation script files:

/h/PDR/Scripts/run_BG_report
/h/PDR/Scripts/run_BH_report
/h/PDR/Scripts/run_BI_report
/h/PDR/Scripts/run_BJ_report
/h/PDR/Scripts/run_D3_report
/h/PDR/Scripts/run_F12A_report
/h/PDR/Scripts/run_F30_report
/h/PDR/Scripts/run_F51_F52_report
/h/PDR/Scripts/run_FF_FE_report

A.2.6.3.1 ReportWriter Specific

The following parameters are required for all reports to define parameters of the job to ORACLE Reports. Where there is a domain of valid parameter values, the parameter name is followed by the legitimate options separated by slashes.

PARAMFORM=YES/NO
Controls display of the runtime parameter form. Set to ‘YES’ when PDR running in DEBUG mode, otherwise ‘NO’.

ORIENTATION=LANDSCAPE/PORTRAIT
All PDR reports except F12A, Reference File Summary, use ‘LANDSCAPE’.

DESTYPE=Screen/File/Printer/Preview/Sysout/Mail
This is set based on user response to the PDR Report Options window. Note that a user selection of ‘Screen’ will result in ‘Preview’ being set, unless PDR is running in a printerless environment, in which case ‘Screen’ is used. PDR does not use ‘Sysout’ or ‘Mail’.

DESNAME=Desname
Contains the output file name when a user selects ‘File’.

DESFORMAT=Desformat
Only applies when ‘File’ is selected. Specifies the characteristics of the printer. PDR uses ‘dflt’.

BATCH=YES/NO
PDR sets to ‘YES’ when destination of ‘Printer’ or ‘File’ is selected. Set to ‘NO’ when ‘Screen’ is selected.

BACKGROUND=YES/NO
PDR sets to ‘YES’ when destination of ‘Printer’ or ‘File’ is selected. Set to ‘NO’ when ‘Screen’ is selected.

PRINTJOB=YES/NO
Set to “NO” for Screen or Printer mode; otherwise set to “YES”. Controls the display of the Reports dialog box that asks whether the user wants to print the report.

A.2.6.3.2 PDR Common Parameters

The following parameters are common to all the PDR reports.

P_USER_ID
User’s ORACLE login ID; part of the key to the records specific to this report request in the collection table.

P_OPLAN_ID
Selected OPLAN_ID for Oplan-based reports; another part of the key to the collection table records specific to this report request.

P_LOGIN_DT
Date and time of this report invocation; another part of the key to the collection table records specific to this report request.

P_CLASSIFICATION
The user-specified classification of the report; it is printed in the top and bottom margins of each page of the report.

P_ZULU_DATE
The time of report generation in GMT format; it is printed on the report.

A.2.6.3.3 PDR Report Specific Parameters

The following parameters are used by the specified PDR reports.

P_SORT_CLAUSE
‘ORDER BY’ clause for reports F11W, F11D, F11E (SQ and TN), BJ, FE.

P_SUPPRESS=YES/NO
Detail suppression option for AMC/Airlift/MSC/Sealift reports. Default = YES.

P_EDIT_LEVEL=A/B/C/ALL
Logical Errors Report - specifies Edit Level setting. Default = ALL.

P_FATAL_ERR=YES/NO
TCC Pre-Edit Report - specifies whether warnings or fatals are to be reports. Default=YES.

P_MODE_TO=POE/POD/DEST
For the F30 Report only. Specified transportation leg.

P_OP_PLN_ID1
For the F52 report - specifies one of the OPLANs to be compared.

P_OP_PLN_ID2
For the F52 report - specifies one of the OPLANs to be compared.

A.2.6.4 ORACLE Kernel

In order to access the ORACLE JOPES Core database for PDR related activities, the user must have an ORACLE account, the user must have permissions to the TABLE_MASTER schema and the JOPES_REPORTS schema, and the user must have synonyms to the JOPES_REPORTS tables. The user account and general database permissions are controlled by the scripts listed below. Refer to Paragraph A.2.2.2.1.6.2.1, PDRSRV Database Segment Installation.

/h/PDRSRV/Scripts/enter_new_oracle_password
/h/PDRSRV/Scripts/determine_oracle_password
/h/PDRSRV/sql/pdr_cr_tablespace.sql
/h/PDRSRV/sql/pdr_cr_jr_user.sql
/h/PDRSRV/sql/pdr_cr_role.sql
/h/PDRSRV/sql/pdr_grant_tm.sql
/h/PDRSRV/sql/pdr_grant_jr.sql

The scripts that control user access to the PDR subsystem are listed below. Refer to Paragraph A.2.2.2.1.6.2.3.1, Enable User for the PDR Subsystem.

/h/PDRSRV/install/pdr_enable_user.csh
/h/PDRSRV/install/pdr_check_user.sql
/h/PDRSRV/install/pdr_assign_to_role.sql
/h/PDRSRV/install/pdr_cr_synonyms.csh
/h/PDRSRV/install/pdr_cr_synonyms.sql

In the Gain PDR application, the interface to ORACLE is established by invoking an RDA function, “RDADBlogin” of the “RDAToolBox” application in the “RDAUtility” library. This function returns a login id that must be used in the creation of SQL queries in Gain scripts.

A.2.6.5 Printer

Currently PDR is configured to print report output to a PostScript printer when the user chooses to print. Printing from ORACLE reports requires that a PostScript printer is defined in ORACLE reports and that it is online. Additionally, PDR requires that the "PDR_PRINTER_DEFINED" environment variable is set. The "set_printer_flag" script is executed by a System Administrator or a DBA to set the variable. If the variable is set to 'NO' then ORACLE Reports output is sent to the screen only.

A.2.6.6 Help Interface

Each of the PDR Gain applications that the user interacts with contains an online Help capability. When the user selects the help function, the screen-specific help is displayed by invoking the LaunchHelp function of the RDAToolBox application of the RDAUtility library. The required parameter to the help function is the name of the text file containing the help information. In addition, there must be a link between the application and the file. This is accomplished by a line in the *help.config* file in the helpdata directory. The PDR textual help files reside in the following directory: */h/PDR/src/helpdata*.

A.3 Compilation/Build Procedures

A.3.1 Required Tools

Building the PDR client segment requires two tools in addition to a UNIX editor: Gain Momentum Version 3.1 and ORACLE Reports 2.0. **Note:** A developer's license is required to build a Gain Momentum runtime.

A.3.2 Building the Gain Momentum Runtime

A PDR runtime must be built to incorporate modifications to any of the Gain Momentum applications that are part of PDR into the production software. A Gain Momentum runtime consists of an executable file named *gain.exe*, application library files that have an extension of "*.glb*," and standard Gain Momentum libraries. The following steps are used to build a PDR runtime using all the applications in the PDR library. **Note**: All files that make up the PDR runtime reside in subdirectories under the */h/PDR/pdr_home/PDRdist* directory.

a. Log in to the PDR development host and start GainMomentum using the following command:

   gmOps &

b. Open the 'Predefined Reports' library.

c. Perform an export on the entire "Predefined Reports" library. From the library menu, select **{File}**, then **{Export}**, and then **{This Folder}**.

d. Exit Gain Momentum by clicking **{File}** and then clicking **{Quit Gain Momentum}**.

e. Log in to the host used to create RDA runtimes, using the authorized RDA account.

f. Set the DISPLAY as appropriate for your display and window manager.

g. Copy the file created in step #3 to a chosen development directory, as appropriate to the JOPES host environment. Currently, the files are copied to a directory created specifically for the new runtime. The new directory resides under the development team's build directory, currently *$PDR_HOME/gainif*. The naming convention for the directory is DDMMYY (DD=date, MM=month, YY=last two digits of the year). underneath our build directory, and copying the files there. This procedure facilitates recreating a build if it becomes necessary.

   Enter the following commands to create the new directory in the correct location:

   ```
   cd $PDR_HOME/gainif
   mkdir <MMDDYY>
   ```

h. Start Gain Momentum using the following command:

   ```
   gmOps &
   ```

i. Open the library "Predefined Reports".

j. Delete each application that will be replaced with a newer version by highlighting the application and clicking **{Edit}** and then **{Delete}** from the menu. These applications will be replaced by those you exported from the "Predefined Reports" library in #3.

k. For each application that was deleted in the previous step, import the new version. To do this click **{File}** and then **{Import}**. This step will create a PDR folder within the PDR library. Open the folder and move the individual applications into the PDR library.

l. For each application in the Predefined Reports library, highlight the application and select File:Check In to check in each of the applications.

m. The application "PDRStartTool" is starts the PDR application and it must be defined as a tool. If there is a Tool menu option in your Gain application, click on "PDRStartTool", then click **{Tool}** and then **{Define Tool}**.

   If there is no Tool menu option, open an application and press **Ctrl-m** to open a message box, then type in the following:

   ```
   defineTool (the "toolcode" of app "RPIStartTool", app "PDRStartTool", app
   "PDRStartTool")
   ```

n. After checking in all the PDR applications, deselect all applications by clicking the mouse in an empty part of the window. Begin building the runtime by selecting File:Make Runtime. This will pop up a separate "Make Runtime" window.

o. In the "Included Tools" section of the "Make Runtime" window, click on the defaults button. The three items that are selected are ACTION-TOOL, DATA-MANAGER, SAMPLE-EDITOR-TOOL.

p. In the “Startup Actions” section there is a “Contents” list box. Highlight “application StartPDR” from that list, then click the “Start With” radio button next to the “Startup Action” label. In the text entry box next to the “Start With” radio button press the arrow button and select “Data Manager.”

q. Press the button labeled **{Non-Sybase RDBMS Access Settings}** and verify that two entries are highlighted (if the development environment does not consist of both an HP and a Sun platform, deselect or delete the reference to the platform that is not present):

ORACLE7	**hp700**	**var/vortex/bin/hp700/gmoracl7.exe**
ORACLE7	**sol2s**	**var/vortex/bin/sol2s/gmoracl7.exe**

r. Press the button labeled **{File System Settings}**.

s. In the “Target Directory” section, replace the “/home/rdart” text entry with “/tmp”.

t. In the “Libraries to Include” section, click on the **{Scan Libraries}** button. It will alert you that this step might take several minutes. Click **{Continue}**. It usually doesn’t take very long.

u. When the scan completes, select the following libraries by highlighting them with the mouse while pressing down the control key:

GCCSUtilities, RDACargo, RDADevTools, RDAForceModules, RDAMain, RDAMapping, RDAMergeCompare, RDAReferenceFiles, RDARequirements, RDASelect, RDATimeLine, RDAUtility.

v. In the “Platforms To Include” section make sure that both the hp700 and sol2s entries are selected by highlighting each while holding the control key.

w. Click **{Make Runtime}**. The system may ask whether you wish to delete the runtime in /tmp if one exists. Click the options necessary to delete any old runtimes in the /tmp directory. It also tells you were a log will be located. The log lets you know whats going on behind the runtime process. If you wish to monitor this log, type: **tail -f <logname>**.

x. A dialog box containing space requirements information will appear. It may say that there is not enough space to build the runtime in /tmp. Ignore the message and click **{Continue}**.

y. If Gain asks “is this the last version before saving?”, answer “no”.

z. After 30-45 minutes, a dialog box should appear saying that the build was successful. Click **{OK}** to clear the box.

aa. Click on **{Dismiss}** to exit the “Make Runtime” window.

bb. Exit Gain Momentum by clicking **{File}** and then clicking **{Quit Gain Momentum}**.

cc. Enter the following command:

```
rm /tmp/runtime/lib/splash.tif /tmp/runtime/lib/sqlhosts.txt
```

dd. Type **exit** to log off from the rdart account .

ee. Change the working directory to the PDR directory where the development group's Gain files reside (currently this is *$PDR_HOME/pdr_home/PDRdist*). Enter the following commands to recursively add write permissions on the data directory and then clear out the directories before copying in the new files:

```
chmod -R +w data
rm -r data
```

ff. Copy the new runtime to this directory.

```
cp -r /tmp/runtime/* .
```

A.3.3 Generating the PDR Report Definition Files

To build the PDR report definition files (indicated by an extension of *.rep*), perform the following procedure:

a. Change the default directory to the */h/PDR/data* directory.

b. Enter the following to initiate the ORACLE Reports Designer (specify the name of the report definition file for which the user wants to generate the binary file):

```
${ORACLE_HOME}/bin/r20desm report_file_name
```

c. From any report writer window, click **{File}** and then click **{Compile}**.

d. From any report writer window, click **{File}** and then click **{Generate}**.

e. Make sure that a filename with an *.rep* extension has been selected in the filename field. Edit the filename so that the name of the directory for the Sun version of the report generator is inserted before the filename: "sol2s/<report_file_name>". Then click **{OK}**. If the *.rep* file exists, the user is asked whether to overwrite it. Answer in the affirmative.

f. Exit ORACLE Reports by clicking **{File}** on the menu bar and then clicking **{Quit}**.

g. The report definition file must also be generated on the Hewlett-Packard. Begin the process by copying the file to the Hewlett-Packard:

```
ftp gccslab8
```

At the username prompt, enter a valid Hewlett-Packard user_id, and at the password prompt, enter the password.

h. Copy the report definition file and exit the ftp utility:

```
put <report_file_name>
quit
```

i. Logon to the Hewlett-Packard using the appropriate account and password.

```
rlogin -l <user_id> gccslab8
```

j. Source the necessary environment variables:

```
source /opt/bin/coraenv
```

k. Repeat steps 2 through 6. Do no add a directory to the filename when step 5 is repeated.

l. End the session on the Hewlett-Packard:

```
exit
```

m. Using the ftp utility, copy the report generation file that was just created back to the Sun and exit the ftp utility. Repeat step 7 and then enter the following commands:

```
get <report_file_name>.rep
quit
```

n. Copy the report generation file to the appropriate directory:

```
mv <report_file_name>.rep ${PDR_HOME}/data/hp700
```

A.4 Modification Procedures

This paragraph describes the procedures that must be followed to modify PDR following receipt of a Government-approved Global System Problem Report (GSPR). The following paragraphs specify information necessary to perform modifications.

A.4.1 Supporting Software

The support facilities, equipment, and software for PDR are listed in Paragraph A.3.1, Required Tools.

A.4.2 Databases/Data Files

Database objects and data files used by PDR and the procedures for modifying them are described in Paragraph A.2.5, Database Variables. PDR nondatabase objects such as scripts, Gain Momentum interface files, and ORACLE Reports report definition files are listed in Paragraph A.1.1, Source Files.

A.4.3 Design, Coding, and Other Conventions

The design and coding standards and procedures in the following paragraphs apply to modification efforts.

A.4.3.1 Coding Standards

PDR was created in accordance with the following coding standards:

a. Language-independent,
b. Gain Momentum,
c. ORACLE Reports, and
d. UNIX scripts.

These standards appear in the JOPES Development Standards (see Section 2.0, item a.) and have been informally provided to the Government.

A.4.3.2 Applicable Procedures

The procedures in the following paragraphs should be used to modify PDR application software.

A.4.3.2.1 Gain Momentum User Interface Software

To update the PDR windows, the programmer must execute the Gain Momentum 3.1 application builder. The programmer must first make sure that the environment variables are set as described in Paragraph A.2.4.1 through Paragraphs A.2.4.3 and A.2.4.5. To modify any of the Gain Momentum scripts for the PDR application, execute the following steps:

a. Logon to the computer on which the PDR Gain Momentum files are available and start Gain Momentum using the following command:

```
gmOps &
```

b. Open the “Predefined Reports” library.

c. Click on the application to be modified.

d. Gain Momentum asks if the application is to be checked out; reply in the affirmative.

e. Make the desired modifications and test the resulting application. **Note:** There is no separate compile/build step for testing because interactive Gain Momentum is based on an interpreter instead of a compiler and any changes can be tested immediately without exiting the Gain Momentum application.

f. When the modifications are successfully completed, click **{File}** from the menu bar and then click **{Check In}**.

g. Follow the steps in Paragraph A.3.2, Building the Gain Momentum Runtime, to build a new runtime.

A.4.3.2.2 ORACLE Reports Report Definition Files

To modify an ORACLE Report, copy the *.rdf* version of the report from the *h/PDR/data* directory. Then enter the following command to execute the ORACLE Reports Designer (substitute the name of the specific report file for "report_file_name"):

${ORACLE_HOME}/bin/r20desm report_file_name

Modify the report as necessary. To run the modified version of the report for test and evaluation purposes, click **{File}** from the menu bar and then click **{Run}**.

After the modifications have been tested, replace the existing version of the report in the *h/PDR/pdr_home/data* directory.

A.4.3.2.3 Script Modification

The location of all PDR scripts is detailed in Paragraphs A.2.2.2.1.1.2, A.2.2.2.1.2.2, A.2.2.2.1.3.2., and throughout Paragraphs A.2.2.2.1.5.3 and A.2.2.2.1.5.4. To modify PDR application SQL scripts or UNIX scripts, copy the desired file from the CM directory hierarchy (currently */export/home/pdr_cm*) and make the desired modifications using any UNIX editor.

A.4.4 Compilation/Build Procedures

The compilation and build procedures defined in Section A.3, Compilation/Build Procedures, should be used.

A.4.5 Integration and Testing Procedures

To perform integration and testing of PDR modification, use the test cases cited in the Software Test Description and Procedures: Pre-Defined Reports (Final) document (see Section 2.0, Reference item e.). Test results should be compared with results recorded in that document.

A.5 Qualification Provisions

The PDR application is delivered on magnetic tape media, using the standard UNIX "tar" format. The current version identification can be verified through manual examination of the "VERSION" file located in the */h/PDR/SegDescrip* directory. This file can be viewed through the use of the procedure described in Paragraph 1.3.